AUTO CARTO 8

Proceedings

Eighth International Symposium on Computer-Assisted Cartography

Baltimore, Maryland
March 29 — April 3

AJK

AUTO CARTO 8

Proceedings

Eighth International Symposium on Computer-Assisted Cartography

Baltimore, Maryland
March 29 — April 3

Edited By
Nicholas R. Chrisman

Sponsored by
American Society for Photogrammetry and Remote Sensing
and American Congress on Surveying and Mapping

COVER PHOTO

Baltimore's Inner Harbor, acquired by the Environmental Protection Agency (EPA) on 22 July 1982 using the EPA-Enviropod camera system.

ISBN 0-937294-88-8

Published by
American Society for Photogrammetry and Remote Sensing
and
American Congress on Surveying and Mapping
210 Little Falls Street
Falls Church, Virginia 22046
USA

Printed in the United States of America

Proceedings

Eighth International Symposium on Computer-Assisted Cartography

AUTO–CARTO 8

Baltimore, Maryland
March 30 - April 2, 1987

FOREWORD

This volume contains the prepublished papers from the Eighth International Symposium on Computer-Assisted Cartography. The printed papers serve the registrants and speakers during the meeting, but they also serve as a permanent record of the state of knowledge in 1987.

This volume contains the overwhelming majority of the papers to be presented at *AUTO–CARTO 8.* Papers are ordered in the volume in the order of presentation during the week, allowing for the overlapping of concurrent sessions. The author index provides alternate access.

The Director and Program Committee of *AUTO–CARTO 8* are deeply grateful to each author and their support staff who made this volume possible. Without their effort, cooperation and attention to detail, the whole could not be assembled from so many parts.

The papers presented in this volume and at the symposium were selected from a large pool of abstracts. The authors of all the 160 abstracts submitted deserve thanks, particularly those whose proposals were not selected. There is a difficult problem involved in selection of papers for an event as diverse as *AUTO–CARTO 8.* To some extent, all persons working in the field should have access to the forum to share their ideas. The conventions operated on that principle turn into massive concurrent programs with no coherence and *espirit.* Over the history of AUTO-CARTOs, there has been increasing attention to a review process to ensure a coherent focus at the event. The Director wishes to thank the members of the Program Committee and others colleagues who assisted in this task. As the papers arrived, the quality seems to equal the standards of any previous *AUTO-CARTO.*

Director
Nicholas R. Chrisman

Program Committee
K. Eric Anderson
Donald F. Cooke
Kenneth J Dueker
Robert W. Marx
Thomas K. Poiker

A NOTE ON THE FUTURE OF AUTO–CARTO

It is gratifying to produce this volume and look forward to the excitement of the symposium at Baltimore. *AUTO–CARTO* events have punctuated my career and the development of this field of endeavor. I am sure that a large number of long-established colleagues and new recruits will join in the intellectual challenge of the Baltimore event.

The seventh symposium and this one have taken roost inside the overall structure of the ACSM-ASPRS Convention. As Director of this event, I wish to thank Mary Clawson, Director of the Convention, and her assistants for their cooperation, assistance and forebearance. The Directors of the Technical Programs of the two organizations, Tom Collins and Phil Mobley, have taken every opportunity to adjust their schedules to accomodate *AUTO-CARTO 8.* They have turned into close friends during the hectic events of the past year. Some colleagues question the wisdom of this association. Some even dream of *AUTO-CARTO* as a budding discipline which will eventually cut its ties with the old ways of doing business. This note is intended to refute that tendency. This symposium should not split off on its own. It generates a special intensity of interest and a high quality of discourse which must be used creatively to renew all of the disciplines that contribute.

The Last *AUTO–CARTO?*

At some point, there must be a last of this series. I personally do not want an honorary place on the podium of *AUTO–CARTO 38* some sixty years from now. Long before then, the interest in a separate symposium on automation in cartography must wane. The core of *AUTO–CARTO* must become the main agenda of the whole convention. I am not sure when to stop the series, but I think it should occur when the whole convention has the sense of energy and the quality of presentations which have become standard at these symposia.

AUTO–CARTO has become a key phrase evoked around the world, and shows the influence of these events. We must try to use the idea creatively instead of falling into the trap of running the symposium out of habit.

Finally, these comments are strictly personal and do not reflect the considered judgement of the sponsoring organizations. The thoughts on future events should not receive undue attention compared to the event at Baltimore.

Nicholas Chrisman
Madison, Wisconsin
20 January 1986

TABLE OF CONTENTS

Basic requirements of digital systems

Models of terrain

Numerical representation for coordinate data

Analytical modelling applications

Models of spatial error

Algorithms for spatial search or query

Generalization and scale changing

TIGER developments

Database architecture

Future directions

Educational software

Applications projects

Expert systems

Vehicle navigation and location

CD ROM

Mixed technical issues (poster session)

Algorithms for cartographic design

Database maintenance

Author index

MULTI-PURPOSE LAND INFORMATION SYSTEMS: TECHNICAL, ECONOMIC AND INSTITUTIONAL ISSUES

Kenneth J. Dueker
Center for Urban Studies
Portland State University
Portland OR 97207

ABSTRACT

Advances in computer, surveying, and mapping technology have had a marked impact on the economic barriers to multipurpose land information systems. This has opened the way for institutional innovations that may help to achieve the data sharing and spatial registration objectives identified in the 1980 National Research Council report on Multipurpose Cadastre.

INTRODUCTION

Since the appearance of the second National Research Council report on the multipurpose cadastre (1983), workers in the field have generally concurred as to the technical soundness of the overall concepts included in their recommendations (Wilcox,1984; Chrisman and Niemann,1985) These include: data layers or themes; the primacy of geodetic control and a base map; and a separate cadastral layer. Unfortunately, the technical aspects of the problem appear to be much more amenable to solution than the economic or institutional ones. Nevertheless, experience shows that advances in the first area can relax and sometimes remove barriers in the other areas. Continuing advances in computer, surveying, and mapping technology have had a marked impact on the economic barriers to multipurpose land information systems. This has opened the way for institutional innovations that may help to achieve some of the data sharing and compatibility objectives identified in the NRC report.

The purpose of this paper is to identify some technical, institutional, and economic aspects of the land records modernization problem: 1) a restatement of the land records modernization problem in terms of technical, institutional, and economic interactions, 2) an assessment of existing approaches to developing the cadastral layer and other map layers, 3) an examinination of promising technology for the development of the cadastral layer and multi-purpose land information systems (LIS), 4) an exploration of LIS implementation, particularly the economic barriers, 5) an examination of opportunities for institutional innovations, and 6) an integration of the findings on technological advances, economic barriers, and institutional innovations into observations on the land records modernization process.

PROBLEM

The troubled condition of land records in the U.S. has received widespread recognition. The National Research Council brought focus to the issue in their 1980 report, "Need for a Multipurpose Cadastre":

> "There is a critical need for a better land information system in the United States to improve land-conveyance procedures, furnish basis for equitable taxation, and provide much-needed information for resource management and environmental planning." (NRC, 1980)

Many agencies collect and use information about land -- its ownership, value, size, location, zoning, natural resources, and use -- in many different forms. Much of that land information, automated or not, is usually maintained by an individual institution for its own specific needs, without knowledge of or concern for its usefulness to others. This results in duplication of effort, higher overall costs, and limited utility and accessibility of the information to other agencies or individual citizens. Nevertheless, Portner and Niemann (1983) have shown that these deficiencies are the result of rational institutional behavior -- when each institution follows its own self-interest. These problems exist because traditional institutional arrangements were developed to meet the needs of a less complex society.

The advance of technology provides opportunities to automate land records processing. Yet, computerization of existing procedures is not, of itself, sufficient. A more "efficient," badly structured system is not what is needed. The whole process needs to be reconsidered and redesigned. In particular, the procedures used to develop assessor's maps need to be examined before it is decided to somehow transform them into digital records. It would seem unproductive to have highly precise copies of inaccurate maps, yet this is exactly what is happening in too many places.

EXISTING APPROACHES TO DEVELOPING THE CADASTRAL LAYER

Kjerne and Dueker (1984) describe two methods for developing the cadastral layer. The first method digitizes points and lines from existing maps, while the other calculates the location of points and lines from deed descriptions and survey measurements. Either of these two methods can be supplemented with base information (planimetry) derived using photogrammetric methods. This information provides visual evidence of roads, fences, *etc.* to aid in placement and fitting of points and lines, and allows the mapper greater confidence as to the location of poorly referenced property boundaries than mapping without such evidence. This kind of evidence, however, can only be supplementary to evidence gathered in the field.

Whether digitized or computed, all the points and lines in the cadastral layer need to be placed into a global spatial framework. Unless control was in place during the period within which the assessor's maps were built and maintained, a major reconstruction process and control densification is necessary to achieve a quality cadastral layer.

The network of control is also needed for spatial reference for other layers so the layers will maintain registration. The spatial registration of resource thematic data to the cadastral layer is an important issue for the management of land and the regulation of land uses. For example, open space planning requires the ability to determine relationships between open space boundaries and boundaries of land ownership in order to identify impacted parcels. Delineated flood plains must be related to the cadastral layer so planners can identify parts of land ownership parcels where buildings cannot be located. Assessors may need to relate the soils layer with the cadastral layer to determine the value of land based on the productivity of agricultural land. These examples show the need for relating the cadastral layer to resource thematic data layers. What are the problems? The 1983 NRC report identified a major problem as map compilation scale differences:

> Resource thematic data such as soils and floodplain boundaries, are normally compiled at map scales between 1:10,000 and 1:100,000. Transferring these already imprecise boundaries, whether by hand or by computer, to a cadastral mapping scale (1:1000 to to 1:5000) implies a higher accuracy than warranted, which may create erroneous information relating to specific parcels of land. (NRC, 1983)

Another problem relates to geodetic control. Separate data compilation scales requires a dense network of control to facilitate adjustment of data from a smaller scale to the larger scale layer. These technical issues affect the ability and opportunity to share data among various organizations. Solving the problems will ease the economic barriers to analysis and create new institutional approaches to the management of land.

NEW APPROACHES TO THE DEVELOPMENT OF THE CADASTRAL LAYER

Land records modernization taking place in a context of single-purpose systems. Major players such as utility companies and public works departments of municipalities are proceeding more rapidly than agencies actually responsible for land records. Rarely are they proceeding in concert. The organizations initiating these developments, however, often have lower spatial accuracy requirements, making their work of little value to those that follow. The "outside plant" spatial accuracy requirements of utilities are largely schematic. Consequently, their land base will not serve other users' needs and utility data will not register spatially to data from other agencies. Utilities refer to it as the "land base" because it is a picture or drawing of street rights-of-way, easements, and key natural features serving as a map base to which to register the distribution system. Generally, each user generates the land base anew as part of initial system setup. The costs of such duplication of effort, as documented by Larson *et al.* (1978), are substantial.

The land base constructed in this manner usually does not distinguish individual ownership parcels. Parcel center coordinates may be related to a parcel identifier relating the graphic object to the non-graphic data file containing parcel attributes. This approach, using parcel centroid coordinates linked to non-graphic data via a unique parcel identifier and a pictorial layer of property lines, predominates as the way to produce the cadastral layer. This method essentially uses the computer to produce a digital equivalent of a new paper map. Although the layer of property lines can be scaled, translated, rotated, and windowed, it is only relatable to other layers in pictorial form.

Figure 1 illustrates data structure options for mapping the cadastral layer. Options 1 through 4 represent the current state of the art, while options 5 through 7 are more powerful extensions. The choice among options will depend on the application. For urban planning and management applications, the parcel centroid usually suffices, while in rural areas with large ownership units and complex natural systems, parcel boundaries are essential. Meanwhile, surveyors need an option that uniquely describes corners and boundaries with the locational rules preserved.

To meet the surveyors's needs within an information system, topologically structured graphic data must be augmented with a record of the reference objects, procedures, and measurement values by which the property boundaries were established. Such a data structure enables a land information system to respond to a greater range of geometric and geographic questions (White, 1984), and is described in greater detail in Dueker and Kjerne (1985) and Kjerne and Dueker (1986).

A topological data structure for the cadastral layer would be expensive, particularly if it would produce only another "picture of a map." If user needs warrant the additional structuring of parcel data in topological form, consideration should be given to recording the evidence discovered by property surveyors so complete reconstruction of a portion of the cadastral map would not be necessary to update locations of cadastral objects.

LIS developments are driven by the need for finer resolutions and the requirement to include ownership considerations in public decisions and plans. The NRC's recommendation (1983) for a multipurpose LIS has achieved a convergence of approaches upon a layered system based on geodetic control and a cadastral layer(Chrisman and Niemann, 1985). A logical extension of the NRC recommendation is the need for a topological data structure that would uniquely record property corners, boundaries, and parcels and the spatial relationships of those objects, and would preserve the rules by which the objects are located. This graphic data structure would then be related to attributes of the parcel by a parcel identifier.

What appears to be a clear case for adopting the most elegant data structure for the cadastral layer is not as clearcut when economics are taken into consideration. As will be illustrated there is no clear strategy or simple solution to the complex technical, economic, and institutional problem of structuring the cadastral layer.

LAND INFORMATION SYSTEM COST ISSUES

The dilemma facing proponents of multipurpose land information systems is economic in nature. The benefits of compatible or spatially registered land data are difficult to identify and to measure. The benefits that can be identified are largely of an "avoided cost" nature (Epstein and Duchesneau, 1984). Additional and new uses of compatible data are largely undefined, though system proponents have "faith" that they will emerge through use of multipurpose land information systems. These new benefits will likely accrue to users of more sophisticated data structures that will allow topological overlays as well as graphic overplotting of layers.

Geodetic control and the resultant accurate base layer necessary to achieve the spatial registration to make data compatible are both expensive. As a result, single purpose systems are being developed, especially by utility companies, to meet their facility management needs, but not their engineering design needs. The schematic representation of their "outside plant" facilities is not relatable on the ground to other utility distribution systems or land features. Apparently, utilities' analyses of benefits and costs indicate that the additional benefits for engineering design do not warrant the additional cost of control. Although the cost to capture map data has been reduced by technological advances in turnkey automated mapping systems, similar cost reduction in control is only now becoming available through technological advances in global positioning systems. This technology, however, has yet to be incorporated into turnkey systems. It will occur as photogrammetric systems are optimized with respect to control densities and direct digital output. These are examples of technological opportunities that will reduce economic and institutional barriers.

The costs, particularly of control, are up-front while the benefits, in terms of avoided costs, lie in the future. It will take a large stream of benefits, discounted, to warrant the large up-front costs. Again, as an example, utility companies have opted to forego the cost of control, which enables them to implement a single purpose system more rapidly. Consequently, they recoup reduced costs faster.

Consortia approaches, as tried in Philadelphia (DVRPC, 1980) have proven difficult in terms of lining up all parties with respect to timing and budget. Questions of control of data also emerge as a major issue. In theory, dividing the cost of the base layer among a number of users makes sense. Accomplishing this has proven difficult for public agencies. Often it is not possible for the public sector to make the investment in data compatibility and share that cost among users. Similarly, utility companies are often precluded by federal or state regulatory agencies from investing in more spatial accuracy than they need, and from establishing an enterprise to sell their base layer.

The institutional and economic barriers are interrelated in terms of differing requirements of agencies and the high cost of additional geographic data detail and spatial accuracy. What may appear to be institutional barriers to a multi-purpose approach may in fact be legitimate differences in need for data detail and spatial accuracy. Similarly, so-called economic barriers may also be different demand functions or willingness to pay for data detail and spatial accuracy.

The economics of land information systems are difficult to assess, partly because many mapping system uses are in a governmental context where it is difficult to arrive at an accounting of costs and benefits. Nevertheless, organizations desiring to invest in a system and data conversion or to upgrade an existing system will want to perform analysis of the economic effects before coming to a final decision.

OPPORTUNITIES FOR INSTITUTIONAL CHANGE

The interrelationship between the economics of land information and the institutions that deal with land information, raise a number of issues concerning barriers and opportunities for institutional change. The principal issues are intergovernmental and interorganizational.

The high implementation cost of new, improved land information systems creates a difficult competition for resources within general-purpose governmental units. Assessors must compete with roads, police and health for resources within county government. Often these other units have problems that are more acute and visible than the assessor's need to modernize land records. Modernization of the mapping function and extension to an interagency multi-purpose LIS is discretionary and difficult to understand and promote. Even though the new system could possibly generate new revenue, this cannot be assured. The cost is more evident than the benefits.

The implementation of new systems will generate land information that will be in demand by other organizations. New mechanisms will be needed for sharing information and allocating the cost of information among organizations.Markets may be gained, lost, or shifted, and organizations will respond to shift by changing strategies and organizational form.

The opportunity for institutional innovations is twofold. One such innovation is the opportunity for public-private partnerships. Another is a growing state role in land records modernization.

Public-Private Partnership

A partnership of federal and state governments, local governments, universities, utility companies, and the private sector is needed. The state must provide progressive leadership, financial incentives, and technical assistance. Local governments and their constituents desire improved information to reduce costs of government. Universities must address important research questions and educate system designers, developers, and users. The universities can also help in facilitating change. Public utilities must be willing to explore longer term benefits of compatible data. If these actors do their job, the private sector vendors will respond to profit opportunities. Creating a viable market for their services may have to be nurtured.

Perhaps the biggest need is for the base layer. A private firm could sell the base layer without certain of the constraints operating on public agencies and regulated utilities. The uncertainty of this market and high capital costs, however, have prevented entrepreneurs from responding to this opportunity. The state, local governments, and utilities must assist in stabilzing this market to provide encouragement to vendors. They can do this by contracting to buy the base layer. However, the same problems that inhibit consortia exist here: it is rarely possible to line up all the clients at one point in time; some agencies or utilities are leading or lagging; budget cycles and needs are not synchronized.

Some doubt whether the private sector is ready to market base layer services, particularly whether the private sector would provide the necessary continuity over the life of a system. Can it easily be turned over to the public sector for maintenance and updating? Possibly some quasi governmental entity, in the form of a regulated monopoly, would be more reliable. A "base layer utility" would insure public control, insure a reasonable return on investment, and would have the same institutional stability as an electric utility.

Growing State Role

Institutional innovations might occur at the grass roots level, reshaping local governments into more effective managers of land information. However, innovation will likely need direction from state governments in order to achieve the desired standardization and compatibility needed for efficient application of technology. Also, an active state role will result from state interest in land related issues, particularly water and natural resource development issues, and transportation and economic development issues. The states will mandate or encourage programs to deal with those issues, and land information will be needed.

With respect to modernizing land records, states have either approached the problem in a broad and comprehensive way or in a narrower problem solving way. Massachusetts and Wisconsin are two states that have addressed the problem of land records in a comprehensive manner, by forming a study commission or committee. The commission in Massachusetts failed to achieve reforms. The comprehensive solution failed to achieve political support for the institutional change and financing of modernization. In Wisconsin, the committee process is still underway and its success in achieving institutional change to effectuate modernization of land records is still an open question.

A number of states have become involved in various forms of modernizing land records through programs of property tax equalization. These programs are often motivated by state school aid formulas requiring that local property tax effort be equal. Under the program, a state agency, usually the Department of Revenue, is empowered to oversee or conduct studies to evaluate the consistency of assessed values to true cash values. State have assumed the role of standardizing assessment practices and have organized the data reported to support those evaluations.

Other states, such as North Carolina and Oregon, have undertaken mapping programs to aid local governments to reconstruct cadastral maps. North Carolina was motivated to improve maps to reduce title conflicts and to ensure equitable taxation of property. Oregon was motivated by the lack, in a number of small counties, of technical expertise to maintain assessor's maps.

CONCLUSIONS AND RECOMMENDATIONS

Technological changes are occurring much more rapidly than institutional ones, reinforcing the tendency to opt for the pragmatic solution of implementing single purpose systems the costs of which can be quickly recovered. There is little incentive to investing in the institutional effort to make systems compatible and share, rather than duplicate, data. Unless the pace of institutional reforms is increased, multipurpose systems will not be achieved. To some extent technological advances will continue to obviate or relax the need for institutional reforms. However, the multipurpose LIS objectives make a purely technological fix insufficient.

The disparity between personal and societal perspectives and behavior forms another consideration. That is, an individual will tend to maximize his personal space and goods and minimize his contribution to the public good. In economic terms this is the "free rider problem" with respect to the provision of public goods. Investment in the multipurpose system is not in the interest of individuals and individual agencies. These short term interests are better met by single purpose systems. This must be offset by new and better institutional and individual incentives.

The technology to provide compatible land information has increased significantly, but institutional innovation has lagged. Institutional innovations in the private sector have been examined for clues for application to the public sector. We find that the public sector will have to act more like the private sector if institutional innovation is to be achieved. There is still considerable confusion surrounding the question of how to approach the modernization of land records. Some view it as primarily a technical problem while others consider it to be primarily institutional. The multipurpose objective both helps and hinders defining the scope of the modernization problem. It raises promises and expectations, but it inhibits the drawing of tight boundaries.

Although institutional and economiç constraints impose significant barriers to its implementation, there has been general acceptance of the NRC's concept of the multipurpose LIS. Institutionally independent and spatially registered layers of data are the key.

REFERENCES

Chrisman, N. and Niemann, B. 1985. "Alternative Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning". Auto Carto 7 Proceedings: Digital Representations of Spatial Knowledge. American Society of Photogrammetry and American Congress of Survey and Mapping, pp. 84-94.

Dueker, K. and Kjerne, D. 1985. "Issues in the Design of a Multipurpose Cadastre". Presented at the 1985 Urban and Regional Information Systems Association Conference.

Epstein, E.P. and Duchesneau, T.J. 1984. "The Use and Value of a Geodetic Reference System". University of Maine, Orono, ME. Available from National Geodetic Information Center, NOAA. Rockville, MD.

Kjerne, D. and Dueker, K. 1984. "Two Approaches to Building the Base Layer for a Computer Assisted Land Records System". URISA 84: Proceedings, Urban and Regional Information Systems Association.

Kjerne, D. and Dueker, K. 1986. "Modeling Cadastral Spatial Relationships Using an Object-Oriented Language'. Proceedings, Symposium on Spatial Data Handling. International Geographical Union.

Larson, B. et al. 1978. "Land Records: The Cost to the Citizen to Maintain the Present Land Information Base. A Case Study of Wisconsin". Department of Administration, State of Wisconsin, Madison.

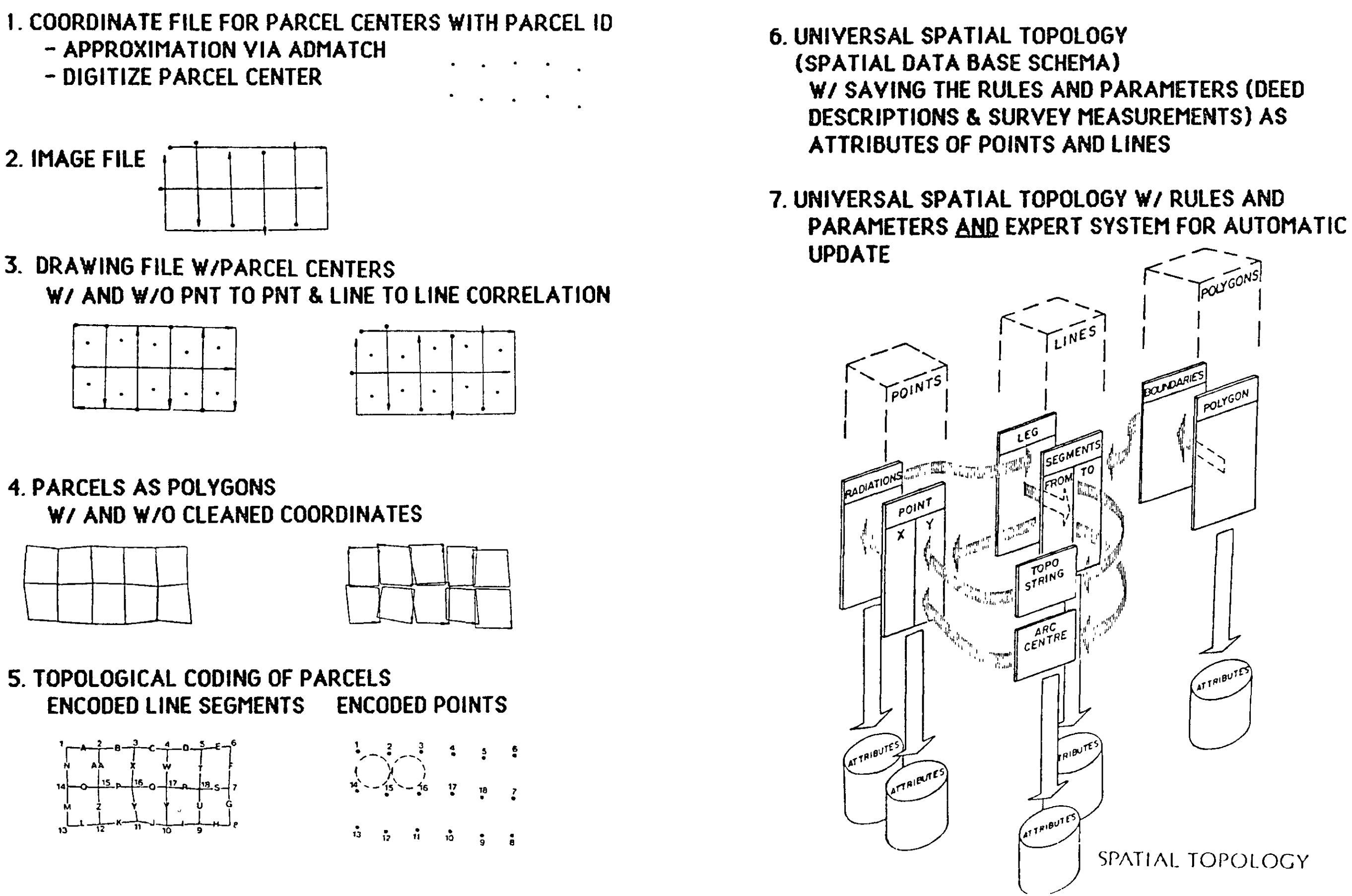
National Research Council Panel on a Multipurpose Cadastre. 198Ø. Need for a Multipurpose Cadastre. Washington, D.C.: National Academy Press.

National Research Council Panel on Multipurpose Cadastre. 1982. Procedures and Standards for a Multipurpose Cadastre. Washington, D.C.: National Academy Press.

Portner, J. and Niemann, B.J. 1983. "Autonomous Behavior: Its Implications to Land Records Modernization". Proceedings of the XV11 Congress de la FIG, Sophia, Commission, Session 7Ø1.3.

White, M. 1984. "Technical Requirements and Standards for a Multipurpose Geographic Data System". The American Cartographer. Vol. 11, No. 1, pp. 15-26.

Wilcox, D. 1984. "Proposal Methods and Procedures for Building a Multipurpose Cadastre Base Map and Cadastral Boundary Overlay". URISA '84. Papers from the Annual Conference of the Urban and Regional Information Systems Association. pp. 223-232.



1. COORDINATE FILE FOR PARCEL CENTERS WITH PARCEL ID
   - APPROXIMATION VIA ADMATCH
   - DIGITIZE PARCEL CENTER

2. IMAGE FILE

3. DRAWING FILE W/PARCEL CENTERS
   W/ AND W/O PNT TO PNT & LINE TO LINE CORRELATION

4. PARCELS AS POLYGONS
   W/ AND W/O CLEANED COORDINATES

5. TOPOLOGICAL CODING OF PARCELS
   ENCODED LINE SEGMENTS   ENCODED POINTS

6. UNIVERSAL SPATIAL TOPOLOGY
   (SPATIAL DATA BASE SCHEMA)
   W/ SAVING THE RULES AND PARAMETERS (DEED DESCRIPTIONS & SURVEY MEASUREMENTS) AS ATTRIBUTES OF POINTS AND LINES

7. UNIVERSAL SPATIAL TOPOLOGY W/ RULES AND PARAMETERS AND EXPERT SYSTEM FOR AUTOMATIC UPDATE

FIGURE 1. CADASTRAL LAYER DATA STRUCTURE OPTIONS

THE USER PROFILE FOR DIGITAL CARTOGRAPHIC DATA

A.R. Boyle
University of Saskatchewan
Saskatoon, Saskatchewan, Canada, S7N 0W0

ABSTRACT

The paper reports on a year long examination of the general user profile for digital cartographic data. The investigation casts doubt on the needs for this data to be in vector form as far as topographic overlays and cadastral boundary lines are concerned. The only actual or proposed used for such data appeared to be as background imagery, and this could equally be met by the much lower cost scan data format. The main question raised for discussion is relative to the policies of the larger digital base map producers. Are the large front-end expenditures of time and money for the production of digital vector data really warranted, or could they more usefully serve the public in a much shorter time period and at a lower cost? The writer believes that the user profile has not been given the attention it deserves. It is hoped that this paper will raise discussion and comments from present and prospective users.

The controversy over raster versus vector digital map data has raged to the writer's knowledge for at least 25 years. Changing situations, user needs and technology require it to be continually reviewed. The writer has just completed a sabbatical year discussing problems with spatial data users and producers in many parts of the world. Some new, although perhaps with hindsight, obvious aspects have appeared and these seem worth discussing at this time.

The most important change is the fact that the 'users' are now starting to make their comments felt; until recently little data except self-generated has been available to make general needs formalized. The previously unstated 'wants and non-wants' now present unexpected viewpoints to data producers who previously thought they knew the user. Unfortunately the question to many data producers of 'what is your user profile?' is met with blank looks or even hostility. Have data producers been spending enormous sums on equipment and hours of work on a product that is not wanted? The large costs involved make continual re-assessment critical.

While the object of this paper is to raise questions and create lively discussion, nevertheless this should be limited to the real points at issue and periodically certain exclusions must be made. The first point of importance is that the main part of this paper is limited to cartography; imagery and GIA will only be discussed later and in no great depth.

Cartographic production departments, when moving to digital mapping, have traditionally preferred to think in terms of the more easily relatable vector format. There is a natural connection with manual digitization where points are pairs of XY coordinates and lines are streams of coordinates pairs. A coordinate resolution adequate to recreate the line graphically at good quality, is used; typical resolutions are 0.001"-0.004".

Many cartographers were disappointed when Automatic Line Followers turned out to be technical and costly failures and they were forced to turn to the less comprehensible precision scanners; however, they still felt the need to convert the scan data to vector format. While scanners of high quality are expensive, nevertheless they are reliable and efficient, making a raster digital copy of a separation sheet in about 20 minutes. Recent discussions with data producers have shown production costs of the vector conversion process to be extremely high; the format conversion is easy, but the problem is the amount of interactive edit required to make precision and program-usable vector data. Times of over 100 hours per sheet are reported; in fact, one producer commented that in comparison with manual digitization times for the same sheet the latter proved to be faster and the equipment costs to be appreciably lower. Digital map data producers have tended to pour money and effort into the creation of vector format data, usually now following the route of scanning and subsequent interactive edit, believing that this meets the demands of the user. However, the front end costs and times have been enormous and it is only recently that the user has been in a position to commment on the usefulness of the produce which has only been gradually provided.

Discussion carried out in 1985 with many digital cartographic data users in the topographic area showed that, at that time and into the foreseeable future, they had little use for the digital data except as a graphic background; in fact no general users of those met could cite any example of general cartographic vector data usage in their computational operations. Until this time they have been using drawn sheets as background but this causes difficulty and they certainly do have a need for a digital graphic presentation so that they can manipulate it for display and plotting; their needs would be adequately and perhaps preferentially met by the cartographic data in raster format. Scan data is still digital and can be manipulated by computers in all the normal ways; it can as easily or even more easily be changed in scale, projection, than vector. It can thus be made to overlay or underlay any other data in vector or raster form and many display systems can handle these two formats simultaneously.

Arguments against the data being in raster format are frequently raised and these must be looked at carefully. The first concern is that automatic data selection can only be done easily on a full separation sheet basis rather than on individual points, lines or polygons; each separation raster image is in a different file. It is only on rare occasions, however, that this is of real importance.

Secondly data is often assumed to be far less compact in raster format than in vector. However, when we examine this we have to appreciate that we are only talking about a raster pixel being black or white (1 or 0) and thus run-length encoding can be used. We can assume that the resolution of the pixel is the same as the coordinate resolution and then only in the case of long straight lines do vectors show an advantage. As soon as the line is curved or irregular then the two formats are relatively similar in bulk. The present rapidly reducing cost of storage makes any difference unimportant in most cases; it is interesting to look at the enormously uncompact storage of vector data to realize that neither producers nor vendors worry about this and even seem to prefer to use excessively large disk and tape storage. The argument of compactness thus has little or no validity.

Cartographers sometimes say that raster data is not as visually 'smooth' as vector, but when the same resolution is used this comment cannot be warranted. In many cases it arises from the fact that for many years coarse grid cells (similar to a large raster pixel) produced this type of presentation.

If scan data were to be agreed as the desired output then two possible forms are possible. Data from scanning may be 'as is' with lines being as wide as they appear on the original separation sheet. The second is to reduce all lines to a standard single pixel width by a line thinning program. This latter software operation takes appreciable cpu time and can lead to a need for interactive edit, because lines can thin so much that they disappear. If high costs and edit times are to be avoided the direct scan data should be produced and used if proved to be acceptable.

It should be remembered that the proposals under consideration are only that data should be supplied to the users directly after scanning and prior to vectorization and intensive interactive edit. If this were acceptable to the user then data could be provided at a much earlier date and at a much lower cost. This is because we now have operational scanners in wide use and do not still depend on manual digitization.

A note on costs gleaned from many agencies over a period of time might be useful. Taking as a basis a reaonably complex small scale separation sheet, eg. for hydrology, the digitization cost range can be anything between $1000 and $10,000. With multiple map series such a difference can have a large financial impact. An efficient manual digitization can often be done for $1000, equipment costs being relatively low. Automatic scanning to produce a raster image need only cost $200; however, to edit this to high quality vector data can cost at least $2000 and frequently much more because of combined operator and equipment costs. If the supply of raster image data alone meets a majority of user requirements ($200) why go to the tediousness of manual digitization or the high costs of editing the raster/vector image data?

It is, of course, not possible to state that all needs could be met by raster data as background imagery. Vector data can be very useful in some cases and essential in others. Technically this does mean that all user interactive stations must be capable of at least visually overlaying raster and vector data. There would seem to be very good arguments for digitizing the culture separation sheets of small scale maps in vector form and storing them in that mode; some boundary sheets may also be treated in this way. However, the vector digitization and storage of parcel boundaries on cadastral maps seems to be best done as raster, because the main use is very much that of imagery only; few people seem to have enough confidence in the boundaries themeslves for exacting computer manipulations. One cadastral data user reported that 94% of his needs were met by the alphanumeric data only, another 5% by the addition of a centroid point and only 1% from the boundary image. Utility mapping is probably best treated as vector digitization and storage, but it is to be hoped that within a few years no digitization of drawn utility maps would be required, the first input being on a display in computer form.

In addition to the fact that input of cartographic data is moving to raster techniques, the user now can have powerful cpu capabilities. This makes it possible to suggest that the usually small amounts of raster data required in vector form (eg. along a transmission line) could be converted by the user. It seems very wasteful to convert all data from raster to vector at high cost because it then might be useful to someone at some time. A number of interactive display systems have a capability of pointing to a pixel and creating a vector line data stream from adjacent pixels; moreover, this can be done at very high speed, the data usually stopping as soon as a junction or line end is reached. In the future a preferable method would be to indicate a vectorization rectangle or polygon and proceed by a batch program.

The question of labelling points and lines must be discussed. Using the raster data image form, any labels on the original will of course be passed through the system to the viewer. If the user carries out vectorization, that user also has to label such lines from the viewed data. Methods have been proposed for automatically labelling node points on a separate overlay and this procedure might be useful in the future.

At this point in the paper we must move from pure cartography to more complex structures such as imagery or Geographic Information Systems. In raster terms this means moving from a simple 2 bit pixel or fine grid cell to one where there may be 256 states or even reams of alphanumeric descriptive information. As would be expected, the data storage increases proportionally and we have to consider the advantages of polygon or raster methodology in a new light. Again we must not be misled by the fact that for many years coarse grid cells were the only method possible, with crude manual digitization and slow manipulation software on the slow computers of that time. We must now consider grid cells as fine as the resolution of the coordinates used in the polygon work, perhaps equivalent to a few centimeters or meters on the ground.

Some years ago the new availability of efficient polygon manipulation changed the capabilities of systems to the extent that they had economic applications in such areas as forestry management. However, not everyone changed to that procedure and the proponents of the fine grid cell as an alternative have a good case. They believe that storage is not appreciably greater and that the grid cell overlay process is superior, particularly when historical, remote sensed or DTM data are concerned. It may be that the polygon methodology has been stretched to its limits and that the next advances will be in fine grid cell work. The phrase 'fine grid cell' must be stressed as many arguments are erroneously based on the noisy visual appearance of coarse grid cells.

Can the complete use of raster data be far distant with the increasing application of scan technology in digitization, edit and plotting, together with the rapidly increasing application of DTMS, DEMs and remote-sensed scan imagery? With the extraordinary increase in memory capacity and cpu speeds both tied to lower costs, the tendency to fine grid cells seems to be there; the economics seem to be in place, the user appears to be pushing in that direction and rapidly improving software is helping.

A major factor in the proposal for more scan data is that it enables both producer and user to proceed towards the future in a step by step way without the initial commitment of enormous funds. Vector digitization may even be the last straw that breaks the back of the camel of traditional cartographic presentation. However, on the other hand it may be that raster digitization is the start of a new type of precision storage cartography because the base material could well be a drawn document, updated manually, and rescanned as needed to produce new data products. This is reminiscent of the RADC developments in the late sixties which were overtaken in the seventies by the explosion of vectorization; perhaps the wheel is turning full circle.

Many of you will know me as a 'vector dedicated' man for many years, even while carrying out appreciable developments in the raster area. I believe the time has now come for me to change my viewpoint to one where data is basically in raster form, but nevertheless not forgetting the great advantages that can be obtained in selected cases from vector formats and being prepared for rapid interchange between the two as the need arises.

OVERLAY PROCESSING IN SPATIAL INFORMATION SYSTEMS

Andrew U. Frank
University of Maine
Department Of Civil Engineering
Surveying Engineering
Orono, ME 04469
FRANK@MECAN1 (Bitnet)

1 ABSTRACT

Combining the information in two or more thematic layers - the overlay operation - is a major problem in geographic information systems.

First a framework for information processing during an overlay operation is given. The geometric intersection operation is separated from the treatment of property or attribute values. In the same way that spatial objects are described by geometric and non-geometric properties, a thematic layer is defined as a geometric partition, dividing space into areas, and the property values associated with each area. To overlay two layers, therefore, requires the computation of the intersection of the two geometric partitions and the combination of the property values.

The calculation of intersections of spatial subdivisions are difficult to execute on computers. The designer of a program to do so must cope with the limited precision of computations on computers. The programs presently available exhibit, at least for some special cases, incorrect behaviour and are computationally demanding.

Two routes seem possible for a spatial information system: either the intersection of two geometric partitions is calculated when it is necessary to respond to a question asked by a user, or all possible intersections are calculated when data is integrated into the spatial information system. The first method seems to have advantages for situations where many existing data sets are combined in a one-time effort to produce a new map. The second appears more suitable for situations where a data collection is built for long-term storage and is intended to be continuously updated.

Separating the geometric operations of intersections from the combinations of the attribute data permits the computation of arbitrary attribute combinations and the classification of attribute values without complex geometric computations.

Work on these concepts was partially supported by a grant from National Science Foundation (grant no. Ist- 8609123) and a joint statistical agreement with the U.S. Bureau of the Census. Contributions from Digital Equipment Corp. are also acknowledged. The Swiss Federal Institute of Technology (ETH), Zurich, and Prof. R. Conzett and his group must be thanked for their hospitality while this paper was prepared. The ideas and opinions expressed here, however, are solely those of the author.

2 INTRODUCTION

A Geographic or Land Information System (GIS and LIS, respectively) or, more generally, any modern spatial information system, contains information related to land. Because of their increased power they are rapidly replacing conventional maps as primary tools for spatial analysis. Overlay techniques, well known from manipulations with conventional maps, are of great importance in many different application areas. [Chrisman 1978]

There exists a danger that we may design tools into the GIS which blindly imitate manual operations. Manual operations usually have some limitations which protect users from the most obvious forms of misuse and abuse. Computer operations, however, are fast enough and have fewer other limitations so that naive users easily produce results with little relevance, and often with completely misleading data.

We present here a theoretical analysis of data processing in an overlay operation. The main result is the separation of geometric operations on partitions and the non-geometric combination of data values in layers for thematic specific property values. An improved understanding is helpful, not only for the design and implementation of lower levels of the GIS, but, more importantly, for the design of the user interface. This leads to systems which are easier to learn and use.

Currently a trend towards database oriented design of information systems can be observed. An information system must be based on a database if users expect timely answers and want to be able to update the system as they learn about changes. Such spatial information systems can form focal points for the organization of land related information in public administration. Organization of data stored for long term usage in a database should follow normalization rules. The computation of geometric intersections when data is entered into these systems fits well into this scheme.

3 SPATIALLY RELATED INFORMATION

In this section we will explore some basic properties of spatially related information. Our position here is only conceptual and no assumptions about an implementation are made. Indeed, this paper strives to separate the theoretical issues from implementation details which, all too often, have made discussions difficult to follow and the results not formulated generally enough for application in other contexts.

Information related to a spatial object either describes the geometry of the object or its spatial or other properties. This separation seems to be trivial, but is of fundamental importance.
spatial object = geometric properties + non-geometric properties

This paper deals only with discrete delimited objects with specific property values. It excludes cases where information is thought of as being related to points in continuous space formulated as a function f(x) (where x is a coordinate tuple describing a point) as, for example, with digital terrain models, or magnetic declination, etc.

Generally, spatial objects may be points, lines or areas, but most overlay processing is carried out on areal objects. We will discuss only these, once again because of limitations in space.

3.1 Non-geometric Properties

Non-geometric information is abstractly a pair consisting of property name and value. The name indicates which property (e.g. land use, land value or height above sea level) is described by the value (e.g. "residential", "$30 per sq. feet" or "350 feet"). It is important to realize that the value alone is not sufficient. The property must also contain an indication of how to interpret the value. A property name can be seen as a function which maps from an object to a value for the named property [Shipman 1981].

A property for which each object has a unique value is called an identifying property and the respective value an identifier or key. Common examples are names, social security numbers, etc.

The values for a property are selected from a domain, e.g. the integer numbers, real numbers, or the names of classes of things. For different properties different encodings are appropriate. Stevens proposed four types: ratio, interval, ordinal and nominal [Stevens 1946]. The selection of an appropriate operation to apply to a property depends on which type of measurement is involved. For instance, it is quite obvious that the calculation of an average is not meaningful for nominal data.

With certain properties we will encounter the need for a null value to represent the absence of a value, either because we do not know it or because it is not applicable. This is not included in the customary algebra (e.g. that which is available on real numbers, etc.) and an extension is necessary. Treatment of null values during operations has been discussed in database literature [Date 1982] [Codd 1986], but a simple solution is not yet known. The widespread usage of 0, -1 or 99 to encode "unknown" can mislead the unwary user and these values seldomly integrate well with operations on the properties.

A systematic study using the new theory of abstract data types or multi-valued algebras [Guttag 1977,1978,1980] [Parnas 1972] to model the categories of encoding measurements would be beneficial. It will show which operations are available on which measurement type and could include the propagation of errors and imprecision. It must specify how "unknown" and "not applicable" are to be treated.

3.2 Identifiers As Locators

The combination of geometric and non-geometric properties need not be simple and direct as assumed above, but can be mediated by describing an object with non-geometric properties and, in lieu of a geometric description, by using a reference which identifies another, geometrically described, object. Such references must be property values which select the designated object uniquely (i.e. an identifier or key value). They can be called locators [Frank 1984]. Most often we use a street address, a parcel identifier, or the name of a town as a locator to describe the spatial object to which some attribute data relate.

Spatial object = identifier for other spatial object (locator) + non-geometric properties

To process the data spatially, we can replace the identifier with the geometric properties of the referenced object and derive a spatial object with an explicit geometric description. It is worth noting that locators can be used in a nested fashion; thus an object can be located using an identifier which references an object which in turn uses a locator to reference another geometric object.

The operation of replacing locators by explicit geometric descriptions can be modelled as a join (exactly equi-join) in a relational calculus [Date 1983] [Ullman 1980]. Please note that no geometric processing is necessary for this operation.

Example:

Relation PARCEL consists of geometric description, parcel-id, valuation. Another relation OWNER consists of parcel-id, owner-name. Assuming that parcel-id is a locator, we can use a join to deduce a combination relation PARCEL-OWNER with geometric description and owner-name.

PARCEL

Id	Value	Descr.
64a-Q	$1025.00	N.E. corner 5th & Elm St.
55e-T	$ 900.50	234 Main St.

OWNER

Id	Owner name
81k-N	Margo Foont
64a_Q	Pelman Twilly
55e-T	ACME Trivet Inc.

Join on Id to produce:
(with appropriate projections on attributes)

PARCEL-OWNER

Id	Owner name	Descr
64a_Q	Pelman Twilly	N.E. corner 5th & Elm St.
55e-T	ACME Trivet Inc.	234 Main St.

3.3 Geometric Descriptions Of Objects

The objects in a spatial information system refer to some objects in the real world for which we know the location and extent in space. The determination of geometric properties of objects is always limited to some approximation and generalization due to limited precision in measurements, and also to limited resolution in the representation, e.g. in a computer system. These approximations pose some special problems which are presently poorly understood and need special attention.

The geometric description of objects may be either points (0-dimensional), lines (1-dimensional), areas

(2-dimensional) or volumes (3-dimensional) - for the context of GIS and LIS we usually exclude the third dimension for information retrieval operations, but modelling using volumes is of great interest for geological and geotechnical applications, including ground water flow modelling, etc. [Carlson 1987].

In this paper we will concentrate on a dimension independent treatment, i.e. results should be valid independent from the number of dimensions used [Giblin 1977] [Frank & Kuhn 1986].

Two basic methods for geometric descriptions are used:

1. vector based, where point positions are fixed with coordinate values and objects are described using lines lines running between these points [Corbett 1975]
2. raster based, where space is divided into a (usually regular) grid and object geometry is recorded as a list of grid cells which approximate the object geometry [Samet 1984]

4 GEOMETRIC CONSIDERATIONS FOR OVERLAY OPERATIONS

In this section we will present some theoretical background for the geometric aspects of the the overlay operation which are independent of the descriptive or attribute data associated with the spatial objects. Specifically we will consider the partition of space into disjoint areas and the intersection of such partitions. All are purely geometric considersations and are independent from the associated values.

4.1 Areas

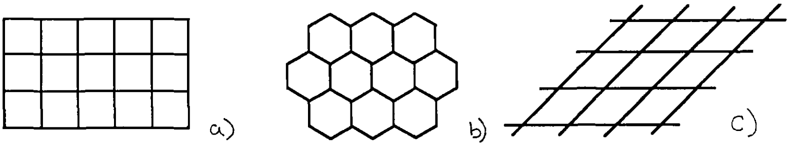
By area we will mean a connected, bounded subset of space (this is different from "regions" in [Tomlin 1983] or the "zones" in [Goodchild 1977]. Area descriptions can be formulated by a bounding polygon or as the aggregation of previously defined areas. Generally the bounding polygon is encoded as a sequence of points which are to be connected by straight lines, but other representations are possible.

4.2 Partitions

Partitions of space are spatial subdivisions (called blocks) constructed so that no two blocks intersect and the sum of all blocks is the total area. Mathematically, partitions are typically created by an equivalence relation, i.e. a relation which is reflexive, symmetric, and transitive (for example: "equal"). All points which have the same property are collected into the same area or block. This reflects the natural assumption that an area designation exists in the first place because its contents somehow have something in common. Partitions can be defined such that all areas with the same value are considered one "region" [Tomlin 1983] or "zone" [Goodchild 1977]; we will only consider connected blocks as areas.

If a spatial subdivision does not form a partition because the areas do not fill the whole space, we can complete it by adding the open space as a defined area (with the property value "null"). If the areas in two spatial subdivisions intersect, a partition is constructed by intersecting all areas and constructing new, smaller areas (which have the same value for all attributes). That is, however, already essentially the solution of an overlay operation.

A special case of a partitioning occurs if the polygons are formed by a regular tesselation of space, i.e. a regular raster, and all areas of interest are described as aggregations of such basic raster cells. It is not necessary that a raster be formed from square fields; other regular tesselations can be used as well (see fig.1) [Diaz 1983] [Samet 1984].



Hierarchies of regular tesselations are commonly used (only a. and c. in fig. 1 form hierarchies). Areas of interest are described as the smallest collection of units from any level. Best known are quad tree structures [Samet 1984] using a hierarchical tesselation based on squares with doubling side length, as shown in figure ... Such hierarchical structures are more compact descriptions for areas than partitions of a single uniform size since they can adapt to differences in required detail [Lauzon & Mark 1984].

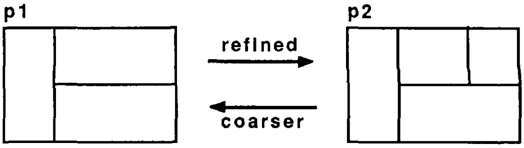
Partitions can be formed by describing each block with a geometric description. Each block can be represented as a closed polygon, but in order to maintain consistency, e.g. with the properties of a partition, a topological data structure is often used [Corbett 1975].

4.3 Partial Ordering Of Partitions Induced By Refinement

A partition, p.2, is said to be a "refinement" of another, p.1, if each block in p.2 is contained in an area of p.1. We can think of a refined partition as the original partition with at least one area subdivided. With this definition of refinement, partitions are partially ordered. Given two partitions, p.a and p.b, it is possible to decide if p.a is a refinement of p.b or p.b a refinement of p.a or neither is a refinement of the other. If both are refinements of the other, they must be equal.

Partial ordering

> a set A (a,b,c...) is partially ordered by a relation greater_than_or_equal_to (>=). This relation is reflexive, antisymmetric, and transitive. Given a and b, either a >= b, or b >= a, or a and b are incomparable such that we can not conclude from 'not(a <= b)' that 'b <= a'



whereas the lattice induced by spatial inclusion is formed by the spatial subdivision in a single block.

4.7 The Most Refined Partition

The lattice of partitions contains a most refined partition (called the infinum) which is more refined than any other partition. The areas or blocks in this partition form the smallest spatial units for which all attribute values are uniform. Such areas have been called Least Common Geographic Units [Chrisman 1975], or Geographic Tabulation Unit Base (GTUB) [Meixler 1985], but the same concept is included in all raster oriented systems where the most refined partition is a regular tesselation, e.g. a square raster. (There is also a least refined partition, which is the undivided universe, called the supremum).

From the lattice structure of partitions with respect to refinement, we know that,

1. if p.0 is a refinement of p.1, and
2. p.0 is a refinement of p.2 (for example, p.0 is the Least Common Geographic Unit partition), then it follows that
3. p.0 is also a refinement of the intersection of p.1 and p.2 [Gill, 1976].

Thus if the partition p.0 is once computed, all geometric intersections can be computed without any geometric operations. Every "coarser" (i.e. less refined) partition is built as set of sets of blocks of the most refined partition. The "geometric" construction of the refined partition becomes purely a set union and intersection operation and no metric operations (like intersection of lines) need be performed. In order to determine the boundaries of the newly formed blocks, a "boundary" operation is applied. [Frank & Kuhn 1986].

The situation where all partitions are constructed from a most refined one is trivially fulfilled with raster representations of areas. Intersection operations in such systems are not difficult. If, however, two partitions must be intersected which are not both formed from the same, more refined partition, geometric constructions to form new, smaller geometric units are required. Such constructions are notoriously difficult.

4.8 Difficulties Of Practical Intersection Computations

Designing programs to compute the geometric intersection of two partitions is difficult and many of the available programs do not properly treat some input configurations. A recent test of several commercially available geometric overlay programs revealed that none worked flawlessly.

First, computers can represent point positions with finite precision only [Chrisman 1984] and this is insufficent for a complete model of geometry. Because of rounding a point may appear to move from the left side of a line to its right by just rotating or scaling [Franklin 1984]. Inserting new points in lines may slightly change the position of a part of the line and perhaps change established topological relations between points and lines.

Secondly, it may be necessary to detect whether a point in one input set is the same as a point in another set (and similarly for the other topological relations). This decision cannot be made by comparing point coordinates only. Because of the inevitable random errors in measurement or a possibly different lineage for computations in the two data sets, coordinates intended to reference the same points may be quite different. Often points within a short distance (tolerance) are identified appropriately, but such methods use some arbitrary threshold which influences the results. If the tolerance selected is too fine, a large number of small areas (gaps and slivers) appear in the result because it is not detected that points in both input sets mean the same point (or a point in one set is incident with a line in the other set). If, however, the tolerance selected is too gross, areas of importance disappear (e.g. roads and rivers). If two points with different coordinates are identified, new, adjusted coordinate values must be selected. Thus points "move" and can come close to other points, with which they are then further identified. This can lead to substantial changes in point coordinates which are not necessarily correct.

These problems are fundamental and due to the statistical nature of coordinate values (coordinate values are non-estimable quantities). A good program should produce, for any consistent input, a consistent output, perhaps with minimal differences dependent on the order of processing. Correct treatment, however, requires additional information to guide the process.

5 TREATMENT OF NON-GEOMETRIC PROPERTIES

After solving the geometric intersection problem we have to combine the associated data. It is important to note that this step is indepentdent from the geometric operations. We have to split the overlay operation into geometric intersection and non-geometric value combination procedures.

5.1 Thematic Layer

Similarly to the composition of a spatial object from geomtric description and non-geometric properties, we define a (thematic) layer as a geometric partition together with the values for a property. The name of the layer is often the property name and a value (possibly "null") must be associated with every area in the partition.

Building layers from spatial objects can change the focus away from the object view.

(thematic) layer = property name + partition + property value for each area (block) in the partition

In lieu of considering single objects, we see all areas with the same value for the property. A region [Tomlin 1983] contains all areas with the same value, but it is not necessarily connected. In order to keep the object view, a layer must contain, as values, the object identifiers, but then only very limited operations are possible on these values. It seems as if the concept of layers and overlay operations abstracts from the "object" characteristic and concentrate on property values.

5.2 Overlay Operation

One of the most important, if not the most important, operation for spatial analysis is overlaying two (or several) thematic layers. The overlay operation can be dissected into several simpler operations:

5.2.1 Intersection Of Partitions - A common refined partition from given input partitions must be found. If a most refined partition has been computed previously, no additional geometric operations are needed.

5.2.2 Value Distribution - The values from the input layers are distributed to each block in the new refined partition such that all new blocks which together form a block from the input layer have the same value. Each new block gets a value from each input set. This step is trivial if the two input partitions are the same, e.g. the same regular raster.

5.2.3 Value Combination - The two (or several) values for each block in the new common partition are combined to form the desired output value. Some typical operations for combining values are: (weighted) average of the two values, thresholds and Boolean combinations.

It is necessary that the operations be legal for the type of encoding used for the measurement (e.g. it is meaningless to add or substract nominally encoded values, even if they are represented with real values).

The customary reliance on standard data types from programming languages (integer or real numbers, character strings, etc.) avoids the issue; such standard types include all necessary operations and more and lead to abuse. Another often used escape, e.g. in [Tomlin 1983] is to translate all values into real numbers so that most customary operations are available. This, however, may encourage users to improperly attempt to perform operation that make no sense, for instance to calculate averages from two different land-use classes (What is the average of INDUSTRIAL, represented as 3.0, and RESIDENTIAL, represented as 1.0? Certainly not AGRICULTURUAL which happens to be represented as 2.0).

5.2.4 Classification Of Results - The values for the areas or block may contain more detail than desired (e.g. the values are represented as real numbers, but the information desired is on an ordinal scale with three values "low", "medium" and "high"). It has been observed that classification on small nominal or ordinal scales is more useful for decision making than apparently more accurate values on ratio or interval scales. It may that the detail of the values implies a precision which is not truely available. It is then useful to reduce the values to a smaller number by some classification method.

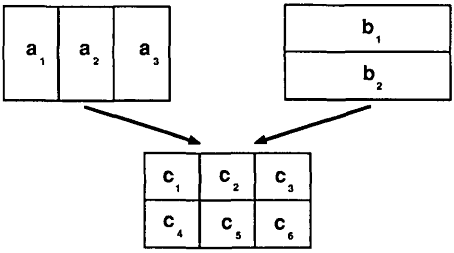
5.2.5 Aggregation - Several connected areas in the resulting partition may have the same resulting value. These areas should be aggregated into one single area. This is not always done, however, often simply because of the difficulty of visually discerning which collection of areas have not yet been aggregated. Nevertheless, aggregation of this kind is necessary for further processing (e.g. to

4.4 Stepwise Refinement Of A Partition

A partitioned space can be refined (with a minimal refinement step) by dividing one block into two new blocks. Similarly a partition can be made less refined (again in a minimal step) by aggregating two adjoining blocks into a single one. With these two simple operations any arbitrary partitioning can be constructed.

4.5 Intersection Of Two Partitions

The intersection operation of two partitions of the same space determines another partition which is a refinement of both. This new partition is the result of the intersection process and can be computed (theoretically) by pairwise intersection of the original areas. The intersecting areas in each original partition can be represented as sets of areas in the new partition.



4.6 Partitions Form A Lattice Structure

As refinement is defined above, partitions form not only a partially ordered set, but a lattice. A lattice is an algebraic structure in which two operations "greatest lower bound" and "least upper bound" are defined for 'any two elements. The results of the operations are unique for elements involved. The least upper bound of two partitions is the least refined partition which is a refinement for each of the given partitions. The greatest lower bound is, analogously, the most refined partition of which both the given partitions are refinements.

From lattice theory we know that, least upper bound is a commutative operation, i.e. intersecting p.1 with p.2 produces the same result as intersecting p.2 with p.1. The associative law is also valid, i.e. for intersecting a number of partitions, it does not matter if we intersect first p.1 with p.2 and then the result with p.3 or start with intersection of p.2 and p.3 and intersect this result with p.1. Finally the least upper bound is also idempotent, i.e. intersecting a partition with itself produces the original partition [Gill 1976]. Unfortunately the actual implementations cannot achieve this theoretical result and results computed may be quite different depending in which order the intersection operations are executed.

The lattice structure formed by partitions with respect to refinement is different from the partial ordering of spatial subdivisions with respect to "inclusion" (e.g. town A contains parcel 218); the latter structure can also be completed to form a lattice [Saalfeld 1985]. The two concepts are related, but distinct: the lattice induced by refinement is one in which the elements are the partitions, permit the determination of the length of the circumference of areas or to recognize the fact that a area of a given value is completely surrounded by areas of another value).

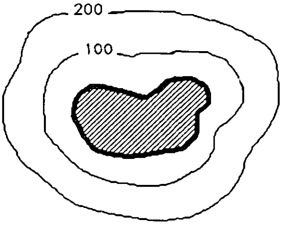
5.3 Application Of Operations On Layers

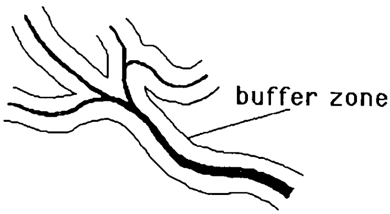
If an operation can be applied to each value in a layer, we can apply the operation to the layer with the meaning that the operation should be applied to each value in the layer (an example is a classification, which maps from real values to some nominal classes). Similarly if an operation can combine two (or more) values from two (or more) layers, we can apply this operation to the layers, again with the semantic that the values referencing the same area should be combined using the given operation (example: computing the average of two or more layers). The results of such operations create a new layer. This is related to the "application" of functions in new functional languages (e.g. Hope [Bailey 1985], or FP [Backus 78] or, with limitations, in APL [Iverson 1962].

Applying operations defined on the values does not include any geometric processing and is usually relatively efficient. Again, it is important to impose the control that only operations which are legal for the given type of values are executed. Such operations, however, are not sufficient for all problems in spatial analysis.

6 CREATION OF LAYERS WITH GEOMETRIC MEANING

For certain operations, overlays with specific values are created, based on geometric operations, e.g. distance from a geometric object.





This may be relative straightforward in a regular raster based system and [Tomlin 1983] gives many examples on how such overlays can be created for irregular partitions. Often a different approach is selected, as when a new partition is constructed which uses boundaries representing specific values for the interested functions (e.g. lines of even 100 foot distances from a pond).

In order to select a method, an effort to understand the information need of the user should be made and then the most appropriate method chosen. In every case, a generalization takes place and some error is introduced; the goal may be to introduce minimal or uniform error to achieve a most equitable result, etc.

7 OVERLAY PROCESSING IN GEOGRAPHIC INFORMATION SYSTEMS

Geographic information processing has in the past been oriented towards mediated batch processing of files. Users would expresse their needs for spatial information and specialists would be employed to produce the information product using the GIS. Increasingly users have been demanding an immediate response, an interactive direct access to the information resource in order to use the available data more effectively and innovatively. Moving from batch oriented, mediated and delayed processing to an interactive situation requires a change in the organization of data.

In order to be effective, the GIS must become easier to use with less effort to learn. Simpler, well structured conceptual models and user friendly interfaces are necessary to reach this goal. It is assumed that the separation into geometric intersection processing and attribute data overlay is useful in this respect.

7.1 Geographic Information Systems Based On Databases

Organization of data becomes important, as the same data is used more often. The database concept where data from multiple sources and with different meanings are logically grouped together and managed by a single software package is appropriate. In advanced systems, multiple users have access to the same data concurrently. Generally, a database management system should contain means to secure data against abuse and accidental loss and it should help to maintain the data consistency.

7.2 Precomputing Intersections

We move from the occasional demand to combine some data files used normally for other purposes to the situation where data are often combined. It becomes thus advantageous to combine and integrate data once and simplify subsequent processing. It may be necessary to secure the help of GIS specialists for the integration, but the users can later retrieve data on their own. For the overlay operation, it is possible to compute the geometric intersection of partitions only once, when data is integrated into the system. The combination of attribute data, however, can not be done ahead of time, as the user's needs are not known. If a spatial data collection is established for long term usage and interactive access for users is demanded, integrating the geometric data and precomputing the intersection of partitions brings the following benefits:

First, the complex and time consuming geometric intersection computations are performed once only. Data integration is generally a time consuming process as consistency checks are made and cleaning of data performed. Additional processing in this phase does not increase efforts significantly. Indeed it may be argued that geometric integration by precomputing the intersections is a necessary part of geometric consistency checking. Eventually, when users pose queries, no additional geometric processing is needed. All overlays can be performed by simply combining the attribute values. A significant saving in computation and in response time must result.

Second, the precomputation of intersections during integration of data opens the possibility to use additional information from the user to resolve dubious cases. We assume that at the time of data integration, users will know more about the quality of the available data. They would beter know the methods used for data collection and the lineage of the data [Chrisman 1983] and are in the best position to select an appropriate tolerances or to decide the identity of points in an interactive dialogue.

7.3 Precomputed Intersection And Data Normalization

Database design uses normalization rules [Date 1983] to design database schema. These rules lead to a systematic break down of data elements (records) to avoid redundancy in various forms. Redundancy is avoided not primarily to reduce the amount of data to store, but to reduce the possibility of inconsistencies and problems during changes (anomaly of update).

Precomputing the geometric intersection can be considered similarly: redundancy is reduced since all common boundaries of all areas are recognized and stored only once. If the data reference the same boundaries often, a situation typically found in large scale geographic data, not only an improvement in performance, but also a reduction in storage may result. More importantly, for any two points or lines the identity problem is resolved and the data processing can use the "unique name" assumption [Reiter 1984].

The overlay of two different layers, both expressed as values with references to the areas of the most refined partition, is primarily a "join" (exactly an "equi-join") using the common reference to the area and efficient methods for execution are known in database literature [Ullman 1980].

8 FUTURE WORK

A number of topics for work have been touched on:

Consistent and formal definition of legal operations for interval, ordinal and nominal data (each extended with values for "unknown", "not applicable", etc.) should be worked on. This would allow for the integration of knowledge about meaningful combinations of values into an GIS and result in advice to the user if inappropriate operations are tried.

Spatial analysis demands more operations than the standard set of arithmetic operations. In [Tomlin 1983] a large number of methods for the creation of rasters filled with values with geometric meaning are given. It would benefit systems which are not raster based to include similar operations (e.g. the "buffer" in ARC/INFO). A systematic study could help to better understand spatial analysis operations and would certainly improve user interfaces and implementations.

Processing of layers does not stress the "object" characteristic, but prefers the "region" approach. Spatial analysis, however, requires both approaches. It is difficult to formlulate queries like: "Find all school districts which contain more than 4 school buildings." In an overlay oriented system. It would be interesting to understand the limitations of each method and see how they can be integrated.

We propose here that the most refined partition is precomputed during the integration of geometric data. This may result in a large set of relatively small areas and spatial objects are then composed of a large number of such small areas. To improve performance, operations on larger objects will be necessary. We assume that application of theories on partially ordered sets and lattices will lead to solutions here.

9 CONCLUSIONS

We have separated a geometric and a non-geometric part in the overlay processing in spatial information systems. Overlay processing is based on layers of data where a given property a value is associated with an delimited area. It is useful to complete the areas to a partition, such that they fill the space completely and do not overlap. Overlaying two layers consists of computing the geometric intersection of the defined areas and of combining the values for the areas.

The geometric operation of intersection of two partitions is difficult to implement on computers and a number of special situations must be dealt with correctly. The problems are due to the limited precision with which point coordinates can be represented. Furthermore, coordinate values for the same point, but from different sources, do not usually agree and identification of similar points in two data sets are difficult. We propose that geometric data be integrated into a system by computing the most refined partition, i.e. compute the intersection of all available data sets. This has the advantage that subsequent processing is simplified. If a GIS is built for a long term usage with interactive access to the data by the users of that data, performing some difficult and time consuming operations like geometric intersection of partitions only once in preparation for quick responses is effective.

Geometric integration of data by computing the intersection can be seen as a form of "normalization" of the data, as common points and lines are recognized and multiple storage reduced. Combining spatial data which is referenced to this most refined partition becomes a non-geometric database join operation.

Operation on single data values can be applied to geographic layers, which contain values for a specific property together with references to the areas in the most refined partition. Such an application of an operation has the meaning that values from the same area are operated on individually and the result again associated with this area. This concept of applying an operation which is defined on a single value to a set of values is similarly found in modern, functional programming languages. It separates the operations on the values from the mechanism necessary for the distribution of the single operation over all the values.

The insight gained by the theoretical analysis of overlay processing is directly applicable for the implementation of new GIS. It improves the design of the system and the coding of its operations. More importantly, the separation into a few relatively simple concepts can benefit the design of the user interface and would result in systems which are easier to learn and easier to use.

10 REFERENCES

Backus, J., Can programming be liberated from the von Neumann style? a functional style and its algebra of programs, Comm. ACM, Vol. 21, No. 8, Aug. 1978

Bailey, R., A Hope tutorial, BYTE Magazine, vol. 10, no. 8, Aug. 1985

Carlson, E., Three dimensional conceptual modeling of subsurface structures, Proc. ASPRS-ACSM Conference, 1987

Chrisman, N.R., On storage of coordinates in geographic information systems, Geo-Processing, Vol. 2, 1984

Chrisman, N.R., Concepts of space as a guide to cartographic data structures, in Dutton, G. (Ed.), First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on Geographic Information Systems, Harvard University, Cambridge, Mass., 1978

Chrisman, N.R., Topological information systems for geographic representation, Proc. AUTO-CARTO 2, Reston, VA, 1975

Chrisman, N.R., The role of quality information in the long-term functioning of a geographic information system, Proc. AUTO-CARTO 6, 1983

Codd, E.F., Missing information (applicable and inapplicable) in relational databases, SIGMOD Record, vol. 15, no. 4., December 1986

Corbett, J.P., Topological principles in cartography, Proc. AUTO-CARTO 2, Reston, VA, 1975

Date, C.J., Null values in databases management, Proc. 2nd British National Conference on Databases, Bristol, England, 1982

Date, C.J., An Introduction to Database Systems, Vol 1, 3rd edition, Addison-Wesley, 1983

Diaz, B.M., Bell, S.B.M., Holroydt, F., Jackson, M.J., Spatially referenced methods of processing raster and vector data, Image and Visual Computing, vol 1, no 4, Nov., 1983

Frank, A.U., Kuhn, W. Cell graphs: a provable correct method for the storage of geometry, 2nd International Symposium on Spatial Data Handling, Seattle, 1986

Frank, A.U., A conceptual framework for land information systems: a first approach, Report 38, Surveying Engr. Dept., University of Maine, 1984

Franklin, W.R., Cartographic errors symptomatic of underlying algebra problems, Proc. Internat'l Symposium on Spatial Data Handling, Zurich, 1984

Giblin, P., Graphs, Surfaces, and Homology, Chapman and Hall, London, 1977

Gill, A., Applied Algebra for the Computer Sciences, Prentice Hall, 1976

Goodchild, M.F., Ross, J.H., Swanson, W.G., PLUS: a conversational regional planning tool, Lands Directorate, Fisheries and Environment Canada, Ottawa, 1977

Guttag, J., Horning, J.J., Formal specification as a design tool, ACM Symposium on Principles of Programming Languages, Las Vegas, 1980

Guttag, J., Abstract data types and the development of data structures, Comm. ACM, June 1977

Guttag, J., et al., The design of data specifications, in: Yeh, R.T. (Ed.), Current Trends in Programming Methodology, Vol. 4, Data Structuring, Prentice-Hall, 1978

Iverson K.E., A Programming Language, Wiley Publishing Co, 1962

Mark, D.M., Lauzon, J.P., Linear quadtrees for geographic information systems, Proc. Internat'l Symposium on Spatial Dta Handling, Zurich, 1984

Meixler, D., Storing, retrieving and maintaining information on geographic strucures, a Geographic Tabulation Unit Base (GTUB) approach, Proc. AUTO-CARTO 7, Washington DC, 1985

Parnas, D.L., A technique for software module specification with examples, Comm. ACM, Vol. 15, No. 5, May 1972

Reiter, R., Towards a logical reconstruction of relational database theory, in: Brodie, M.L., et al. (Eds), On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Springer Verlag, New York 1984

Saalfeld, A.J., Lattice structures in geometry, Proc. AUTO-CARTO 7, Washington DC, 1985

Samet, H., The quadtree and related hierarchical data structures, Computing Surveys, Vol. 16, No. 2, June 1984

Shipman, D.W., The functional data model and the data language DAPLEX, ACM Transactions on Database Systems, Vol. 6, No. 1, March 1981

Stevens, S.S., On the theory of scales of measurement, Science Magazine, vol. 103, 1946

Tomlin, C.D., Digital cartographic modeling techniques in environmental planning, Ph.D Thesis, Yale Univ., 1983

Ullman, J.D., Principles of Database Systems, Computer Science Press, Potomac MD, 1980

FUNDAMENTAL PRINCIPLES OF GEOGRAPHIC INFORMATION SYSTEMS

Nicholas R. Chrisman
Department of Landscape Architecture, 25 Ag Hall
University of Wisconsin–Madison, Madison, WI 53706
BITNET: CHRISMAN@WISCMACC

ABSTRACT

The primary goals in GIS design to date have been focused on technical efficiency. The fundamental principles for an information system do not derive from pure laws of geometry or from computing theory, because they must reflect the basic goals of society. While social goals may seem nebulous, they can be described adequately for resolving some of the basic technical choices. Certain fundamentals can be determined by digging deeper into the reasons behind an information system. Important social functions lead to **mandates** that provide the impetus for **custodian** agencies. Even more fundamentally, geographic information systems should be developed on the primary principle that they will ensure a fairer treatment of all those affected by the use of the information (**equity**). Certain solutions, though efficient in their use of computing do not support the effective use of institutions or the equitable results of the analysis.

WHAT IS A FUNDAMENTAL PRINCIPLE?

GIS has come of age. Over the past twenty years, those inside the community have marvelled each year at the expanding sophistication and power of our tools. Success and expansion are nice, but dangerous. Those who have built the tools know how fragile they are, and particularly how fragile our fundamental understanding. Some of the current success is achieved by exploiting the easy parts of the problems. The tough issues, temporarily swept under the rug, will reemerge, perhaps to discredit the whole process.

This article has a presumptuous title for anything of the length of a proceedings paper. However, as Director of this Symposium, I felt it important to discuss the fundamentals because they may be obscured if the papers concentrate exclusively on specific technical developments.

A number of recent symposia on research needs have emphasized the lack of fundamental theory for GIS and related fields (Smith, 1983; Onsrud and others, 1985). Each report calls for more theory, but without specific suggestions. The field of GIS involves some components, such as knowledge engineering or Geo-Positioning Satellites, that are emerging technologies in the joyful chaos of discovery. The field also involves some of the oldest sciences and professions, such as geometry and land surveying that trace origins back for millennia. It is hard to invent a geometric problem for modern computer displays which was not drawn with a stick on the

Athenian sand three thousand years ago. Any gaps in geometric theory were filled during the eighteenth and nineteenth centuries when a series of great geometers generalized the field far beyond the rudimentary needs of a GIS. This essay will attempt to provide a partial answer to the quest for basic theory for GIS, but the result may be different from the intentions of the above reports. This essay will not enumerate principles in the abstract, but will concentrate on those useful in illuminating the choices behind an overall data model.

AUTO-CARTO is about computers and what they are doing to alter the polyglot disciplines that address spatial information. The primary issues at this symposium are technical ones, since our technology is still far from complete. Technical development requires choices between competing alternatives, and many of these alternatives are completely unexplored in a field like ours. In early exploratory research it is fine to try out a hunch, but as the field matures there is a need to develop more formal and consistent principles to guide the selection. Also, as our technology finds its way into practical use, it must be accountable economically, but also politically, socially and even ethically. The principles developed in this paper are "fundamental" because they try to address the deep issues of why we collect and process geographic information.

To provide a focus for an essay constrained to the proceedings limits, I will focus on the principles that apply most directly to the "data structure" debate, perhaps better known as "raster versus vector". This paper is derived from a seventeen year excursion in automated mapping, but the principles that I now see as fundamental are not the ones that I have expounded at earlier AUTO-CARTOs. At the first two events I presented papers as a partisan of vectors (Chrisman, 1974; 1975). By 1977, I had decided that the argument between rasters and vectors involved such different concepts of space that the proponents could not share the frame of reference required for a true debate (Chrisman, 1978). At that point of development, the debate was thoroughly theoretical, since no complete system had yet been developed. Like many others, I put my energy into building a real system, hoping to resolve the issues by direct demonstration, not theoretical argument. Now that essentially complete systems exist, the debate should be reexamined.

THE DATA STRUCTURE DILEMMA

Throughout the development of GIS, there has been a competition between data models – a sign of vigor, but also a sign of confusion. For the purposes of this paper, I need not be more specific than three basic alternative models: raster, CAD (originally Computer-Aided Design, but now a term of its own) and topological. The raster model prescribes the geometric elements as cells in an integer space. Epistemologically, this model ties back to the atomic theory of Democritus and the modern inheritors of that approach, such as Ernst Mach (1906). Using the simplicity of enumerating objects in the integer space, many measurements can be treated on a common reference. The raster approach has many proponents, but most of the arguments are based on technical considerations (Peuquet, 1979). The two "vector" models adopt a geometry of continuous space (the model of Aristotle and his successors) to position points, lines and areas. The CAD model places the primitive objects into separate "layers", but does not introduce any further data structure. The topological model takes the same primitive objects, but places them into a network of relationships.

These data models were originally driven by technology. The reason for the grid cell was simplicity of programming, and the related raster pixel was determined by the simplicity of hardware designs for remote sensing. Similarly, the vector approach reduced complex graphics to tractable primitives. At one time, vector devices competed with the raster ones. On the hardware front, the technological gap has vanished. Virtually all "vector" devices use raster displays, including advanced page-description devices like the LaserWriter that printed this paper. In most cases, however, there is still a distinction between the raster and vector levels of implementation that highlights the continuing conceptual gap.

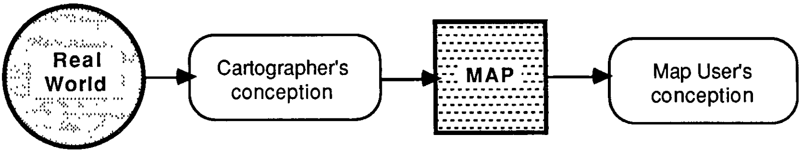
Because the roots of the "debate" are epistemological, there is no chance that the issue will vanish (Chrisman, 1978). There is a need to develop another path to describe the fundamentals of geographic information systems. This paper will attempt to produce these fundamentals from aspects of human society, then to demonstrate their lessons for the data model debate.

Requirements Studies

There have been a variety of procedures used to justify a particular design for a GIS. Many systems are built as experiments in technology, then are promoted without consideration of alternatives. This deplorable phase has to be expected in the early development of any field. A useful theory of GIS would provide a guide to the appropriate data structure and other characteristics of a system from some more fundamental basis. At the moment, the most prevalent approach is a "user needs assessment" or "requirements study" which provides an approximation to system design through a social survey approach. Similar to a time-and-motion study in industrial engineering, an analyst assembles a description of what is currently done, tabulates the results, then formulates a system to replace the current process. In many respects, the current state of affairs is the appropriate basis for a decision, but it introduces certain limitations. On one extreme, it may simply automate chaos without understanding it or improving it. More typically, the promise of the analysis is a more "rational" system. It seems to be an item of faith that every organization must treat information as a "corporate" resource. The analysis is looking for duplicated effort and redundant information. Yet, these irrationalities all arose for quite specific reasons. Often these reasons relate to the differences between choices which are rational in the narrow framework of a given agency, but irrational from a more global perspective (Portner and Niemann, 1983). The reasons behind the "irrational" components of the *status quo* are likely to be potential causes of failure. Hence a *user needs analysis* has limits in doing too little (replicating a bad system) or doing too much (using system analysis to overrule institutional arrangements and foster political infighting). Another approach is required to develop a theory to cover the whole problem.

THE CARTOGRAPHIC COMMUNICATION MODEL

One candidate for a theory is the cartographic communication model. This became a central tenet of academic cartography (Robinson and Pechenik, 1976) during the same period that GIS was developing. The communication model attempts to deal with the role of maps as visual communication with some elaboration of the general scheme presented by the diagram below.

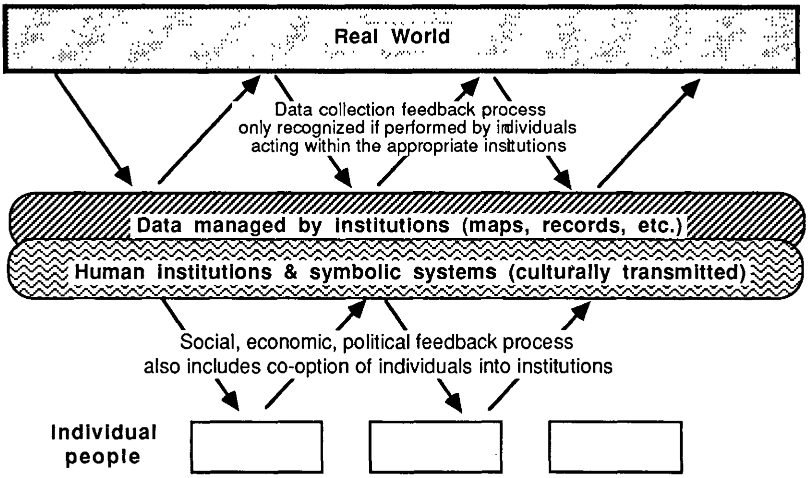


Schematic outline of cartographic communication model

This model provides a mechanism to understand the role of a master artist-cartographer, like Erwin Raisz, who created a whole style of maps to communicate his ideas about the landscape and the processes that formed it. Beyond this rare case, the model is less help. The basic communication model offers little help to understanding non-academic cartography, even those in its manual form. Few manual cartographers have design control over the series that they produce. Maps are defined by a system of conventions and standards that have developed over many centuries.

To understand the purpose of a modern geographic information system, the role of the map is an inadequate guide. The map product serves as a visual channel of communication, but it must be interpreted inside its frame of reference to impart meaning. Most communication models recognize the role of a frame of reference – a common system of symbols, values and interpretations. The system of these beliefs comprise the complex that is called **culture** by anthropologists. While the communication model places the individual person in the key role of sending or receiving messages, the cultural frame of reference exists without an explanation. From the anthropological point of view, culture exists and is transmitted through procedures of acculturation where individuals in a society learn roles, symbols and interpretations. In some cases, the culture is all-encompassing, but there can be substantial diversity in the package of beliefs that a particular individual receives. For example, the term *culture* might conjure up the idea of a simple society of hunter–gatherers with a unitary set of beliefs shared by all. The modern anthropologists would be the first to point out a less unitary reality even in the simplest societies. In our modern society, culture is fragmentary and subdivided. To apply the cultural perspective to spatial data handling, disciplines (geographers, cartographers, etc.) and guilds (lawyers, property surveyors, etc.) represent groups that maintain their identity over time. The individuals recruited into the group are trained to adopt the shared system of values. This explains, for example, how plat maps can be so uniform across the country without direct communication amongst the county agencies that do the work. The sense of what a plat map should look like is transmitted through the discipline and persists no matter which person does the drafting. In short, the person has little control over the data content. The long-lasting and culturally transmitted structure of disciplines is more central than the issues of perception.

In the context of a GIS, the communication model must be modified to accept a cultural kind of transmission. The diagram below is an attempt to present the coherence of most processes over time.


Real World
Data collection feedback process
only recognized if performed by individuals
acting within the appropriate institutions
Data managed by institutions (maps, records, etc.)
Human institutions & symbolic systems (culturally transmitted)
Social, economic, political feedback process
also includes co-option of individuals into institutions
Individual
people

The diagram, though much more complex than the original communication model, does not portray the existence of many competing disciplines and institutions. These distinct units are unlikely to share the same goals and directions, leading essentially to a multidimensional diagram. It is important to refine the notion of culture as it applies to information systems, for example, Hardesty (1986) provides a useful overview of one aspect, cultural adaptation, for the geographic audience. However, for the purposes of this essay, a nuanced theory is not crucial.

It is important to understand the primary motivation for the collection and distribution of spatial information. The existence of a given guild or discipline in a society is not fore-ordained, despite the convention blather common in any social group that claims a central role for themselves in the universe. We can discount as cute or presumptuous a society whose name translates as "THE People", but we tend not to apply this filter to the statements of disciplinarians who claim GIS as their exclusive preserve (references deliberately excluded). There must be some larger motivation that can help clarify and adjudicate.

TOWARDS ANOTHER MODEL

The first conclusion of the cultural argument is that geographic information is a human, social commodity. It is not strictly empirical and objective. This conclusion is dangerous. If taken too far, there is no consensus, and all opinions are equally valid. Fortunately, though each human perception may vary, they are submerged in a cultural system which cannot permit such extremes of relativism. Social structures provide the basic framework of meaning for geographic information.

A set of fundamental principles cannot attempt to be universal. This essay applies most specifically to the polity of Wisconsin, but it applies fairly closely to other states of the US and the provinces of Canada. The general principles presented are transportable to societies that share the same European roots, with adjustment for divergences of legal, political or social systems. This argument will actually apply less to a corporate geographic information system (such as one maintained by a forest products company) than it will inside a socialist planned economy.

A second observation is that geographic information systems are not new at all. Some kind of system has functioned for centuries. The new technology offers many improvements of efficiency, speed and analytical accuity; I do not mimimize these advantages. The new technology disrupts many of the constraints that determine the structure of the old system, which makes it odd that *user needs assessment* is a common path for systems design. The current way of doing things is a useful guide, but perhaps it shows some features of a social and institutional nature often ignored in the systems design approach.

For example, the analysis of the existing system in a municipality will uncover terrific duplication of parcel base maps of varying vintages. The modern technologist, quite rightly, wants to sweep the slate clean and adopt the more rational "normalized" approach to data where only a single true copy is maintained. Technically, this approach is defensible and necessary. Unfortunately, most system design stops with the facts of data management, ignoring the reasons behind the duplication.

Mandates

The important data collection functions of society are not carried out for technical reasons. The creation of property maps, zoning maps and all the other municipal functions are not driven by a benefit/cost ratio. Each record is collected and maintained in response to a social need as expressed in the legal and political system. The search should not be for the flow of data, but for the **mandates** that cause the flow. A mandate, which may be a law, an administrative rule, or even simply a customary practice, provides the definitions of the objects of interest along with the procedures for processing and implications for use of information. In place of the social survey approach, mandates provide a deeper view of why information is collected by certain actors. The legal library may be a better guide to what is intended. Of course, every society has some dissonance between the formal rules expressed in laws and the rules that actually govern conduct. In some societies this gap can seem large to an outsider (when the issue of bribery comes up). Still, the gap is a predictable part of a cultural system. In the case of North America, the rule of law is a very major cultural value, and consequently, the gap should be miminized.

A case of duplication. There is a need to distinguish types of duplication using an example drawn from experience in Dane County, Wisconsin

(Sullivan and others, 1985). On the surface, there is a clear-cut case of duplication of parcel maps involving the County Surveyor and the Zoning Administrator. The zoning mandate does not include an authority over parcels, but parcels are depicted to show the zoning so that the citizens can interpret zoning relative to their holdings. Originally, the zoning maps were made by copying the parcel maps (quick and easy), but since then all changes to parcels have been drafted independently on the two copies. For some particular reason long forgotten, the zoning maps happen to be the only ones that record the tax identification numbers used to collect the county's main source of revenue. The Surveyor's parcel map attempts to portray both the locations described in textual information recorded by the Register of Deeds and the spatial units used in the official tax list. There is not a direct correspondence of these definitions, because they derive from independent mandates. The manual system attempts to handle both needs, but imperfectly. A system which merely removes the duplication between zoning and surveyor will miss the real problem of two groups with independent mandates to define ownership parcels.

Mandates, as formal rules, are implemented by people acting inside institutions. In addition to the external mandates, any institution develops its own internal rules. Some of these are disciplinary, because the people share a common basis of training and language. Sometimes professional ethics can override the mandate or divert its intention. For example, although the property records are maintained for the citizens, the banks effectively require the citizen to use a lawyer or title company to perform the work.

The people inside institutions are important to consider, particularly if a new technology threatens their system of values. As Stein Bie (1984) pointed out at AUTO-CARTO 6, we have to construct systems that serve more goals than simple technical efficiency. His point concerned the personal satisfaction of the workers, but it should be extended.

Custodians

Mandates lead to institutions that carry them out. These institutions have a strong stake in self-preservation, which the agents of technical change might easily interpret (in a surprisingly self-centered point of view) as opposition to progress. There is another solution by recognizing that certain institutions, through their mandates, are **custodians** of their particular records. Instead of opposing progress, the modernized system could become their prime agenda. The modern system provides a much better mechanism for an agency to carry out its fundamental charge.

EQUALITY AND EQUITY

Both the concepts of mandate and custodian were presented, in initial form, as part of the "Institutional Reasoning" for a GIS presented at AUTO-CARTO 7 (Chrisman and Niemann, 1985), but they were not tied to the underlying goals of society. That paper called for a balance between technical reasoning and institutional concerns in the design of data bases, as if the two were equal. Technical efficiency is measured most commonly as the ratio of benefits to costs. Some recent work (Bernhardsen and

Tvietdal, 1986) claims remarkably high ratios, presumably with the idea of influencing public judgement. However, many potential projects have favorable ratios of benefits and costs. The actual decisions taken rely on other principles. There are many possibile principles that transcend technical efficiency, but the most important one to geographic information systems is a complex involving equality and equity.

Equality and equity derive from the same root, and may be easily confused. However, in social science usage, these terms have developed two usefully distinct meanings. Equality refers to rights and other concepts which are allocated to all citizens identically. By contrast, equity is used to refer to a less absolute sense of fairness. Social concerns for both equality and equity are fundamental. Each political philosophy essentially derives from a different operational definition of one or the other. A properly acculturated person will insist that their particular system follows obvious logic, while others are curious aberrations.

The American political system places great value on certain political rights, distributed with strict equality. However, the system of equality does not extend into economic matters very far at all (in contrast to socialist theories). The American capitalist system is founded on the principle that economic production depends on *inequality* to provide incentives to promote efficiency. Okun (1975) describes the unavoidable conflict in his monograph *Equality and Efficiency: The Big Tradeoff.* Okun characterizes many of the political battles in American life as quarrels over the distinction between the rights which are equally shared and the economic goods which are unequally distributed for reasons of efficiency. The barrier is never absolute. For example, no mattter what the theory of equal political importance, the rich can effectively buy greater access to decisionmakers. Also, society will not tolerate the pure capitalist markets that would leave some unfortunates literally to starve.

In the field of geographic information systems, certain activities require strict equality. Rights of access to information must be universal or they are too easy to abuse. However, the line between equality and efficiency is not as difficult as it is in the general economy. The less stringent issue of equity becomes quite crucial. It is through the concept of equity that society tries to deal with the unequal distribution fairly. As a simple example, a strictly equal tax (each citizen pays the same) is not equitable, since the citizens have different economic means. Throughout the country, one prime political issue about land is equity in property taxation. Hence, any information system that deals with property will not be judged simply on its technical performance, but on its contribution to equity.

From my experience with local governments, officials are cautious at first and quite concerned about public expenditure. The initial argument must be one of strict benefit/cost and economic efficiency. However, when the full analytical power of the automated system is available, the results are used to ensure fair treatment that could not be quantified as benefits.

PRESCRIPTIONS

To carry out the principles presented above, these general rules apply:

Geographic information must be collected and managed by public agencies that have a long term stake in the process, not some *ad hoc* central group. The test of such an agency is the **mandate** that provides definitions, quality standards, and other characteristics.

An information system should be organized on a decentralized model that acknowledges the independent mandates of the contributing agencies. One approach is to declare a **custodian** agency for each element of the whole information system.

Finally, **equity** appears to be a more important goal than technical efficiency and benefit/cost ratios. Geographic information systems should be developed on the primary principle that they will ensure a fairer treatment of all those affected by the use of the information.

IMPLICATIONS FOR DATA STRUCTURES

Technical alternatives for GIS should derive from these principles. Certain technical solutions, though efficient in their use of computing do not support the effective use of institutions or the equitable results of the analysis. The most crucial decision is the issue of a basic unit of analysis. Any system of arbitrary units, whether raster pixels, quadtrees, or map tiles, imposes a technical construct onto the objects defined by statutory mandate. Society does not define property in convenient regular rows and columns for easy programming. Similarly, natural processes do not limit themselves to mathematically neat descriptions. It may be possible for software to use these technical tricks at lower levels which are isolated from the user level, but the operations should not degrade the integrity of the definitions.

The bulk of definitions implied in mandates fit into the general vector model of points, lines and areas. Topology is a natural part of many systems such as common law, not the abstruse extra that the newly converted sales reps describe. The human eye/mind combination is so used to association by contiguity, that the uninformed cannot believe that the CAD computer knows nothing about adjacency.

Technical concerns may argue for monolithic, completely overlaid databases, of the form propounded as GEOGRAF (Chrisman, 1975) and now implemented for example as TIGRIS and TIGER. My argument for GEOGRAF was flawed because it centralizes definitions. It substitutes technical efficiency for the logic of mandates and displaces authority away from the custodian agencies to programmers much less aware of the requirements. The search for technical efficiency must not be allowed to overturn political choices without careful examination through the political process. The true challenge is to use the increased sophistication of our automated systems to promote equity and other social ends which will never fit into a benefit/cost reckoning. I am convinced that the future of geographic information systems will lie in placing our technical concerns in their proper place, as serious issues worthy of careful attention. These technical concerns must remain secondary to the social goals that they serve.

ACKNOWLEDGMENTS

The ideas expressed were developed during the Dane County Land Records Project which involved close collaboration with many including Ben Niemann, Jerry Sullivan, Kate Beard and many others. Partial funding came from USDA Hatch grants. This paper provides a personal interpretation, not necessarily the conclusions of all participants.

REFERENCES

Berhardsen, T and Tveitdal, S. 1986 Community benefit of digital spatial information: *Proceedings AUTO-CARTO London,* Vol. 2, 1-4

Bie, Stein 1984, Organizational needs for technological advancement: *Cartographica,* Vol. 21, 44-50

Chrisman, Nicholas 1974, The impact of data structure on geographic information processing: *Proceedings AUTO-CARTO I,* 165-177

Chrisman, Nicholas 1975, Topological data structures for geographic representation: *Proceedings AUTO-CARTO II,* 346-351

Chrisman, Nicholas 1978, Concepts of space as a guide to cartographic data structures: in Vol. 5 *Harvard Papers on Geographic Information Systems,* Reading MA, Addison Wesley

Chrisman, Nicholas and Niemann, Bernard Jr. 1985, Alternative routes to a multipurpose cadastre: merging institutional and technical reasoning: *Proceedings AUTO-CARTO 7,* 84-94.

Hardesty, Donald 1986, Rethinking cultural adaptation: *Professional Geographer,* Vol. 38, 11-18

Mach, Ernst 1906, *Space & Geometry in the light of physiological, psychological and physical inquiry,* Chicago, Open Court (reprinted 1960) translated by Thomas McCormack from articles in *The Monist*

Okun, Arthur 1975, *Equality and efficiency: the big tradeoff,* Washington DC, Brookings Institution

Onsrud, Harlan, Clapp, James, and McLaughlin, John 1985, *A Report of the Workshop on Fundamental Research Needs in Surveying, Mapping, and Land Information Systems,* Blacksburg VA, Virginia Polytechnic

Peuquet, Donna 1979, Raster processing: an alternative approach to automated cartographic data handling: *American Cartographer,* Vol. 6, 129-139

Portner, James, and Niemann, Bernard J. Jr. 1983, Belief differences among land records officials and users: implications for land records modernization: *Proc. URISA,* 121-135

Robinson, Arthur, and Petchenik, Barbara 1976, *The nature of maps: essays towards understanding maps and mapping,* Chicago, U. Chicago Press

Smith, Lowell K ed. 1983, *Final Report of a Conference on the Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development,* Redlands CA, Environmental Systems Research Institute

Sullivan, Jerome, Niemann, B., Chrisman, N., Moyer, D., Vonderohe, A., Mezera, D. 1985, Institutional reform before automation: the foundation for modernizing land records systems: *Proceedings ACSM,* 116-125

AN ADAPTIVE METHODOLOGY FOR AUTOMATED RELIEF GENERALIZATION

Robert Weibel

Dept. of Geography
University of Zurich
Winterthurerstrasse 190
CH-8057 Zürich (Switzerland)
K491170@CZHRZU1A.EARN

ABSTRACT

Digital Elevation Models (DEM) are used in a wide range of applications. Several institutions worldwide are now involved in the collection of DEM data. DEMs are generally compiled at large scales (e.g. 1:25,000) with high accuracy. However, users frequently require data at different scales and for differing purposes (e.g. analysis, display etc.). To derive models at reduced scales, a generalization process has to be applied.

This paper describes the development of an adaptive methodology for automated generalization of DEM data. The principal feature of this methodology is the adaptive selection of a suitable generalization method according to the scale of the resulting map and the characteristics of the given terrain: For smooth relief or minor scale reductions, a collection of filtering techniques (global and selective) is applied. For rougher terrain or larger scale reductions, a heuristic generalization procedure is used which works directly on the basis of structure lines.

INTRODUCTION

Digital Elevation Models (DEM) are increasingly becoming important as a source for geographical analysis and digital mapping. Possible application areas include terrain analysis (e.g. slope, aspect, visibility) for erosion studies, hydrology and various planning purposes, and the derivation of various display products (e.g. contour lines, shaded relief) within digital mapping systems (Burrough 1986). More and more institutions involved in the study and mapping of the earth's surface are collecting DEM data. The compilation generally takes place at a relatively large scale (e.g. 1:25,000) to achieve maximum resolution.

However, users frequently require data at various scales and for different purposes. If relief display is to take place on a smaller scale than the DEM was originally compiled, some of the details in the original data have to be eliminated: The original DEM has to be generalized. Since we are in an automated environment, this data reduction should be carried out automatically but in a cartographically consistent manner. This stands in contrast to the resampling processes used in data reduction for analysis purposes: Resampling for data reduction is guided by statistical criteria and not by visual effectiveness. The generalization procedure should take account of the purpose of the resulting data or map, and of the characteristics of the given terrain. Major scale reductions should be possible.

The development of a methodology for automated cartographic generalization of DEM data was aimed at, given the general frame that the procedures should

- run as automatically as possible, with a minimum of subsequent adjustements;
- perform a broad range of scale changes (from large to small scale);
- be adaptable to the given relief characteristics and to the purpose of the resulting data or map (selection of the best-suited method);
- provide the opportunity for feature displacement based on the recognition of the major topographic features and individual landforms (for major scale reductions);

- work directly on the basis of the DEM;
- enable an analysis of the results.

Several approaches already exist in the field of automated relief generalization, but even the most promising ones do not fulfill all the criteria stated above. Generalization of contour lines fails to address landforms individually and thus allows no major scale reductions. DEM filtering (Loon 1978, Zoraster 1984) applies a global filter operator which does not pay attention to local terrain characteristics and only smooths the data, which again allows only minor scale reductions. Information-oriented DEM filtering (Gottschalk 1972) is locally adaptive but is also restricted to simplification and elimination of details. Heuristic approaches which are based on a generalization of the terrain's structure lines (Wu 1982, Yoeli 1987) are promising for the treatment of rough relief or for major scale changes, but will be hard to operate in low or undulating relief without any clear structure lines; furthermore, they are still very much in an experimental stage and need to be refined.

It is clear that no single strategy can achieve all the above-stated requirements and can cover all scale ranges and all possible applications. The methodology we will describe in the following is therefore composed of a collection of generalization procedures, each one to be suited for a specific sub-area of relief generalization.

GENERALIZATION PRINCIPLES AND OPERATIONS

Cartographic generalization is carried out by applying various generalization operations to the original map. We can identify four basic operations which are, however, not clearly separable (Fig. 1):

- eliminate (select)
- simplify
- combine
- displace

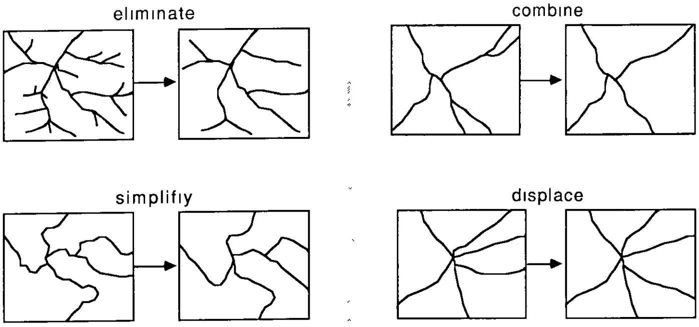


Fig. 1: Basic generalization operations

We do not want to come up with yet another definition of generalization operations or processes (cf. Steward 1974), but use this classification to clarify our understanding of the process of cartographic generalization:

"Eliminate" and "simplify" are conceptually and algorithmically relatively simple. They do not cause major locational changes of the features processed, and are therefore without severe impact on neighbouring elements. They always address features individually (in a clearly separable way) and can therefore be applied in a sequential manner. While they may remove details from the original data they do not create new features or structures.

"Combine" and "displace" are of higher complexity. They involve positional transformations which have an effect on neighbouring features. Because displacement of one element may cause a chain reaction of relocation of other features, they cannot address individual elements sequentially, but they rather do it in a parallel way.

They reorganize available space and build up new structures (e.g. combinations and placeholders). "Combine" and "displace" require a great deal of knowledge of the character and shape of individual features as well as processing strategies which record the spatial interrelationships of features and proceed in a synoptic manner.

The application of the specific generalization operations is determined by various criteria. The most important ones are: 1) scale reduction, 2) map or data complexity, and 3) generalization purpose.

Scale reduction: The smaller the scale of the resulting map or data, the less space is available for the individual map elements. Elimination and simplification does not solve this problem. A reorganization of map space has to take place, calling for combination and displacement.

Data complexity: The more complex the original map or data, the more likely will features interfere if scale is reduced. Here also feature combination and displacement has to be applied.

Purpose: In a digital environment, the purpose of generalization can take new form. In addition to just creating new maps from old ones, a user may want to transform a digital cartographic data base into a different yet generalized data base in order to save storage or processing time in subsequent manipulations, or for other processing purposes. In these cases the central issue is to diminuish the data contents while changing the geometry of the data as little as possible. It results in a mere data reduction, i.e. elimination and simplification. This process is equivalent to a generalization caused by minor scale changes.

Considering the above points, we can distinguish between two kinds of generalization procedures of digital data, according to the types of basic generalization operations applied:

- Filtering: Only elimination and simplification are used. Filtering may be applied only for minor scale changes and data of low graphical complexity. It is also used for controlled data reduction.
- Generalization (*): These procedures are oriented towards graphical output and involve all four basic generalization operations mentioned above. They are used for major scale reductions and/or data of high graphic complexity.

This range of requirements must be kept in mind when developing a strategy for automated generalization of any cartographic feature.

METHODOLOGY OUTLINE

We propose a methodology for automated relief generalization based on the observations stated in the previous section. We are currently implementing this methodology for gridded DEMs, but it could be modified to operate on other types of DEMs such as the TIN model (Triangulated Irregular Network).

This methodology provides for an adaptive selection of appropriate generalization procedures according to the conditions under which the generalization process takes place: scale reduction, relief characteristics, and generalization purpose. This is done by branching into one of two sub-processes, the fitering sub-process or the generalization sub-process (see Fig. 2 for methodology outline). The selection is made either by an operator who applies a priori knowledge, or by means of a selection procedure (i.e. global characterization of the relief type). The selection is guided by the perception of relief character through statistical measures (simple statistics of height distribution, local height changes, slope variation, fractal dimension, texture parameters).

For minor scale reductions and/or relatively smooth relief, a filtering procedure (either global or selective) is applied. For rougher topography or major scale changes, a heuristic generalization approach is taken. It works on the basis of the relief's structure lines (i.e. valleys, ridges, and other breaklines), assuming that these lines are geomorphologically meaningful. The various generalization operations are applied to the structure lines, and after this step the resulting gridded surface is reconstructed through interpolation.

(*) The term "generalization", in this particular case, is to be understood as a subset of the entire process of cartographic generalization. "Filtering" is another subset.

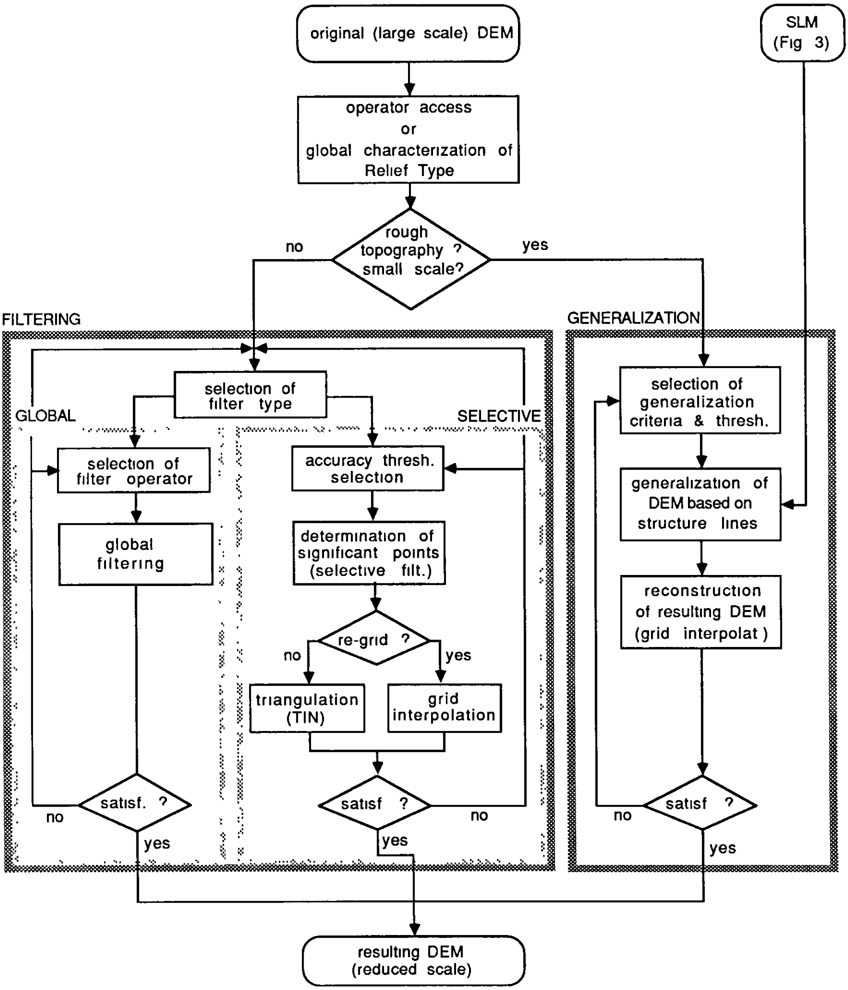

original (large scale) DEM
SLM (Fig 3)
operator access or global characterization of Relief Type
rough topography ? small scale?
no
yes
FILTERING
GENERALIZATION
selection of filter type
GLOBAL
SELECTIVE
selection of filter operator
global filtering
accuracy thresh. selection
determination of significant points (selective filt.)
re-grid ?
no
yes
triangulation (TIN)
grid interpolation
satisf. ?
no
yes
satisf ?
no
yes
selection of generalization criteria & thresh.
generalization of DEM based on structure lines
reconstruction of resulting DEM (grid interpolat.)
satisf ?
no
yes
resulting DEM (reduced scale)

Fig. 2: Outline of generalization methodology

FILTERING SUB-PROCESS

Filtering procedures can be applied to automated relief generalization if scale reduction is modest and/or if the given terrain is relatively smooth.
Two filter types are used in our methodology: 1) global filtering, and 2) selective filtering. The choice between the two alternatives is made under operator control.

Global Filtering

Basics: This filter process operates globally; it does not adapt to local relief features. However, it can be applied if scale reduction is only minor (depending on relief complexity). Global filtering is computationally the least expensive of all generalization procedures in our methodology. It is equivalent to position invariant two-dimensional filters in image processing. Global filtering can take place in the spatial or in the frequency domain.
Operation: For the time being, the selection of a suitable filter operator is guided by an operating person but it would also be possible to control it via the previously made global characterization of relief type. A selection among several filter operators is given. It is also possible to concatenate low and high pass filters for edge enhancement.
The resulting filtered DEM can be viewed through shaded relief display. If the result is not satisfactory, action can be resumed to re-select another filter type or filter operator.

Selective Filtering

Basics: Selective filtering is more sensitive to local variations of terrain in that it is guided by the information content of the individual data points (i.e. it is position variant). The basic idea is to select data points with high significance and to drop the ones with low information content. The approach is related to Gottschalk's filtering of TINs (Gottschalk 1972); however, his solution was computationally inefficient. In our methodology it is also applied to gridded DEM. Because it is sensitive to local relief features, selective filtering can cover a wider scale range and rougher relief than global filtering. Moreover, because it seeks to eliminate only insignificant or redundant points, it can be used as a means for controlled data reduction (e.g. grid-to-TIN conversion).
Operation: After an accuracy threshold has been selected, the set of points is triangulated and each point in turn is temporarily deleted, and an estimated elevation value is interpolated at its position. This action is iteratively applied to all points, and after each iteration, the point with the least difference between actual and estimated elevation value is definitely eliminated. This is done until all points with a difference less than the selected threshold have been eliminated. The amount of accuracy in the resulting data (i.e. the information content) can hereby be controlled.
This procedure can only work acceptably fast, if the re-triangulation caused by point elimination and the interpolation of estimated elevations can be computed locally (i.e. only among neighbouring points). Algorithmically, this task is complex. Development of a differential algorithm for local adjustment of triangulation is now under way at our institution (Heller 1986a). To test our methodology, we are using a pragmatic approach at the moment. It identifies points along structure lines and subsequently selects further points based on their difference to neighbouring cells.
After the set of significant points has been determined, the operator selects the further processing steps. If he desires data reduction only, he will choose a TIN structure as the resulting DEM; if he wants the original DEM to be generalized by selective filtering, the resulting generalized grid is reconstructed through interpolation of significant points. In either case, control returns if the shaded relief display shows no satisfactory result.

GENERALIZATION SUB-PROCESS

If substantial scale reductions and complex topography have to be handled, mere feature elimination and simplification processes are not sufficient; we have to combine and displace relief features and reorganize available space (e.g. smaller landforms have to be combined into larger landforms). These new structures have to reflect the original character of the topography (Imhof 1982).
Structure lines (i.e. valleys, ridges, and other surface-specific edges) build the structural skeleton of the relief. In manual cartography, they are used to support the generalization process. For small scale generalization and in rough topography, we therefore rely on the structure lines.

The flow of the generalization sub-process is as follows (Fig. 2): The structure line model (SLM) serves as the basis for the heuristic generalization procedure. It holds the geometry, topology, and feature attributes of the structure lines of a particular DEM. After setting up this model (which needs to be done only once in an initialization step), the structure lines are subjected to generalization processes (elimination, simplification, combination, displacement), and the new skeleton serves as input to the reconstruction of the resulting gridded DEM at a reduced scale. As in the filtering sub-process, the result can be visualized through shaded relief display. If it is not satisfactory, the generalization can be re-started with different generalization criteria.

Generation of SLM

The structure line model (SLM) is generated by the extraction of the geometry of the structure lines, their concatenation to form network topology, and a subsequent classification of the landforms (see Fig. 3).

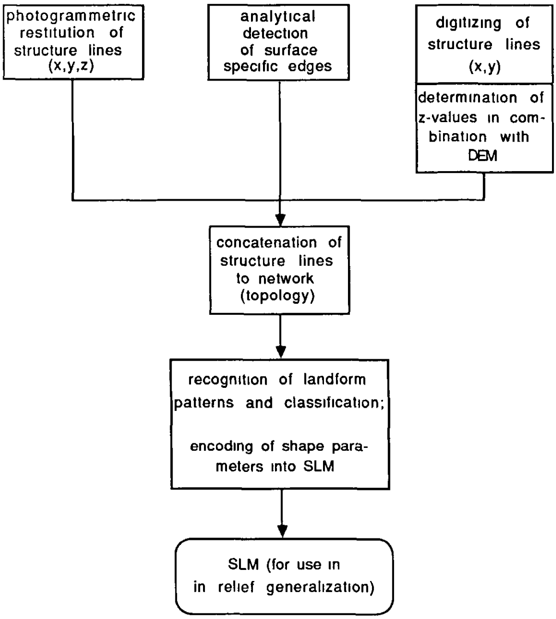

photogrammetric restitution of structure lines (x,y,z)
analytical detection of surface specific edges
digitizing of structure lines (x,y)
determination of z-values in combination with DEM
concatenation of structure lines to network (topology)
recognition of landform patterns and classification;
encoding of shape parameters into SLM
SLM (for use in in relief generalization)

Fig. 3: Generation of structure line model (SLM)

Geometry and topology of structure lines: The aim of the SLM is not only to record valley and ridge lines (which are the main features of fluvial relief), but also other prominent edges (e.g. edge lines of glacially eroded valleys). The sources for geometry are: 1) photogrammetric restitution (Makarovic 1976); 2) digitization (and subsequent z-value interpolation); and 3) automated or analytical detection. The quality of the structure lines influences the subsequent generalization process.

If structure lines are to be found analytically, the procedure used is a combination of heuristic approaches (for valleys and ridges) and image processing algorithms (for other egdes). If certain edges are already known (e.g. from photogrammetry), they are no longer searched for. Edge detection is followed by a vectorization step to concatenate the surface-specific points into connected line systems. This task is not trivial especially if edges are not clearly pronounced. Analytical detection still needs some subsequent interactive editing.

If the geometry is determined through photogrammetry or digitizing, the individual structure lines can be topologically connected through operator interaction or through a topology building process.
Landform classification: To guide the generalization of the structure lines, some information on the importance of these local structures is required. The individual structure lines must therefore be classified according to their prominence. Possible parameters are: edge length; stream order (for valleys); average height (for ridges); volume of pertinent land orm (ridges); ratio volume / area of landform (ridges). This information can be used to form a generalization hierarchy.
Again, if the edges are determined through photogrammetry or digitizing, the operator can assist in that he visually sets an attribute of relative and absolute importance for each edge.
SLM formation: For each structure line, its geometry (x,y,z), network topology, and landform parameters (as attributes) are stored. The SLM data are pertinent to the original data and as such are permanent. The SLM generation needs to be done only once.

Generalization Operations

The generalization of the structure lines can be exemplified by the above-mentioned basic generalization operations:

Eliminate: Based on previously selected thresholds and on attribute information of the SLM. This operation is applied first.
Simplify: Second operation. Simplification of the course of the remaining structure lines according to selected criteria by means of known line simplification algorithms (see Zoraster 1984 for a discussion).
Combine and displace: The simplified edges are now subjected to combination and displacement, controlled by the landform information of the SLM. Algorithms to be developed can profit of experience with automated feature displacement in name placement and line generalization (Zoraster 1984 for further references).

Only a few general guidelines for the application of the individual generalization operations are known from manual practice (Imhof 1982, Zoraster 1984). Our aim is to gradually and iteratively develop heuristic rules that could control generalization. This would also give further insight into formalization of fuzzy cartographic knowledge.

Relief Reconstruction

After the network of structure lines has been modified by generalization, the resulting gridded DEM can be reconstructed through interpolation. The interpolation procedure should generate smooth slopes between the edges, but it should not destroy the breaks. An interpolation base on triangulation and bivariate quintic interpolation is used in our case (Heller 1986b).

CONCLUSION

We have proposed an adaptive and comprehensive methodology for automated relief generalization which combines different approaches. It serves as a comprehensive framework for testing of existing procedures and for the development of new ones. We have shown that it is most important to adapt automated procedures to the degree of scale reduction to be made, to the complexity of the given terrain, and to the purpose for which the generalization is applied. To meet these requirements, we have set up two alternative strategies: A filtering sub-process and a generalization sub-process. For most of the filtering procedures and for some of the generalization procedures, tentative solutions have been implemented. These algorithms can now be tested and refined, and the knowledge for control structures (e.g. thresholds, generalization rules etc.) can be developed. Future research has to address the following points:

- development of hard criteria to select either filtering or generalization procedures, based on statistical relief characterization and on the amount of scale reduction;
- improvement of analytical detection of structure lines;
- develoment of adequate measures for local landform classification;
- formalization of knowledge from manual generalization practice;
- development of appropriate feature displacement algorithms.

ACKNOWLEDGEMENTS

The work for this paper was supported by a grant of the Swiss National Science Foundation. The author wishes to thank Prof. Kurt Brassel, PD Dr. Haruko Kishimoto, and especially Mr. Martin Heller for valuable discussions and for reviewing the manuscript.

REFERENCES

Burrough, P.A. (1986): *"Principles of Geographical Information Systems for Land Resources Assessment"*, Monographs on Soil and Resources Survey No. 12, Oxford: Oxford University Press, 193 pp.

Gottschalk, H.-J. (1972): "Die Generalisierung von Isolinien als Ergebnis der Generalisierung von Flächen", *Zeitschrift für Vermessungswesen,* Vol. 97, No. 11, pp. 489-494.

Heller, M. (1986a): *"Computergestützte Modellierung geologischer Strukturen"*, Internal Report, Dept. of Geography, University of Zurich.

Heller, M. (1986b): "Interaktive Modellierung in der Geologie", *Karlsruher Geowissenschaftliche Schriften,* Series A, Vol. 4, pp. 173-183.

Imhof, E. (1982): *"Cartographic Relief Presentation"*, Berlin: deGruyter, 425 pp.

Loon, J.C. (1978): *"Cartographic Generalization of Digital Terrain Models"*, Dissertation Paper, Ann Arbor: University Microfilms International, 199 pp.

Makarovic, B. (1976): *"A Digital Terrain Model System"*, ITC-Journal, 1976-1, pp. 57-83.

Steward, H.J. (1974): "Cartographic Generalization: Some Concepts and Explanation", *Cartographica Monograph,* No. 10, 77 pp.

Wu, H.H. (1981): "Prinzip und Methode der automatischen Generalisierung der Reliefformen", *Nachrichten aus dem Karten- und Vermessungswesen,* Series 1, Vol. 85, pp. 163-174.

Yoeli, P. (1987): *"A Suggestion of a Methodology for a Computer Assisted Generalisation of Topographical Relief"*, (in preparation).

Zoraster, S., D. Davis, and M. Hugus (1984): "Manual and Automated Line Generalization and Feature Displacement", *ETL-Report ETL-0359,* 184 pp.

SYSTEMATIC SELECTION OF VERY IMPORTANT POINTS (VIP) FROM DIGITAL TERRAIN MODEL FOR CONSTRUCTING TRIANGULAR IRREGULAR NETWORKS

Zi-Tan Chen and J. Armando Guevara

Environmental Systems Research Institute
380 New York Street
Redlands, CA. 92373

ABSTRACT

Selection of a set of significant points from a raster digital terrain model is important for constructing a triangular irregular network. The set of points should contain information of terrain surface as rich as possible.

INTRODUCTION

The most common form of digital terrain model is the raster data structure. Its dense grids represent terrain surface very well for some applications. However, another new data structure has obtained more and more attention recently.

For representation of terrain, an efficient alternative structure to dense raster grids is the Triangular Irregular Network (TIN), which represents a surface as a set of non-overlapping contiguous triangular facets, of irregular size and shape.

The TIN data structure shows a better solution to overcoming problems caused by the non-stationary property of the terrain surface. Also, for some applications, such as shading, cataloging, and visibility, TIN has a nice implementation.

TIN can be directly generated from random point data. However, for constructing TIN from raster DTM, it is not so simple. First, DTM usually has too many pixels which can not all be selected for constructing TIN. Second, if one uses all pixels to construct a TIN, some advantages of TIN, such as simplification and generalization, are lost. Only a subset of pixels from the total pixels can be used for the generation of TIN. Thus, a key question is raised: 'Which pixel should be selected ? ' and 'Which pixel can be ignored?'. The principle here is that, between two pixels, the more important point should be selected.

'VIP' PROCEDURE

Before answering the question of which are the more important points among all pixels, a significance of each pixel must be evaluated. Here the significance of a pixel means how great a contribution the pixel can make to the representation of the surface. Our goal is constructing a triangular irregular network (TIN) to represent the original terrain surface by using the least number of points. We have to select some pixels, while other points have to be thrown away. When we say a pixel P1 is more important than another pixel P2, we mean that a more precise triangular irregular network (TIN) can be constructed if we use the pixel P1 in the set instead of pixel P2. The function of the VIP procedure is to select a set of pixels. The set must have two properties:

(1) For certain precision, a TIN constructed from the set has the least number of points than any other set.

(2) Among all possible sets, with the same number of points selected from the DTM, the set can construct the most precise TIN than any other set.

In other words, any point belonging to the selected set should be more important than any point that does not belong to the set.

Evaluation of significance of a pixel

We have to know how important a pixel is before selection of pixels. The VIP procedure calculates the significant degree of each pixel. An improved spatial high-pass filter is used to produce this significant degree.

High-pass filtering

A picture or an image can be represented in either spatial domain or frequency domain. In frequency domain, low-frequency (long wavelength) components represent major features on the original picture, such as overall skelton, major spatial distribution, etc. High-frequency (short wavelength) components represent detail features, such as edges, peaks or pits. These properties of spatial filtering have been used in digital image processing widely. For example, high-pass filters can do edge enhancement for images to find features.

High-pass filters can also be used to select significant feature points from digital terrain surface models. A pixel should be selected only if we can not predict its value from its neighbor pixels. For example, if a pixel has an average value from its eight neighbors, this pixel is not important enough to be selected. In other words, the significance of a pixel can be evaluated by measuring its changing behavior from its neighbors. This measure can be done by high-pass filters, such as spatial differential or a Laplacian operator. For terrain surfaces, in our applications, an improved spatial differential high-pass filter is used.

In the one-dimension case, the second order differential of a function

$$Y=F(X)$$

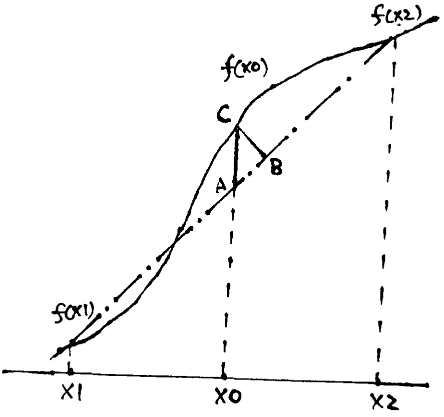
can be noted as:

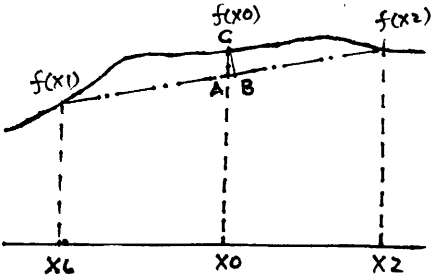
$$\begin{aligned} d^2Y/ dX^2 &= F''(X) \\ &= 2 * [F(X0) - 0.5 * (F(X1) + F(X2))] \\ &= 2 * [F(X0) - A] \end{aligned}$$

The distance AC, shown in Figure 1, can be used to measure the behavior of change.

Improvements

The first improvement is an enhancement of the change measure. The change is measured by distance BC instead of distance AC. The consideration behind this improvement is an effort for better distribution of significant values of pixels. Comparing two examples in flat areas and steep slope areas, we can see why measuring BC is better than measuring AC. Distance BC is actually reflecting the real offset of change, especially in the steep slope area. At a given direction, a significance for the pixel is evaluated by measuring the distance BC

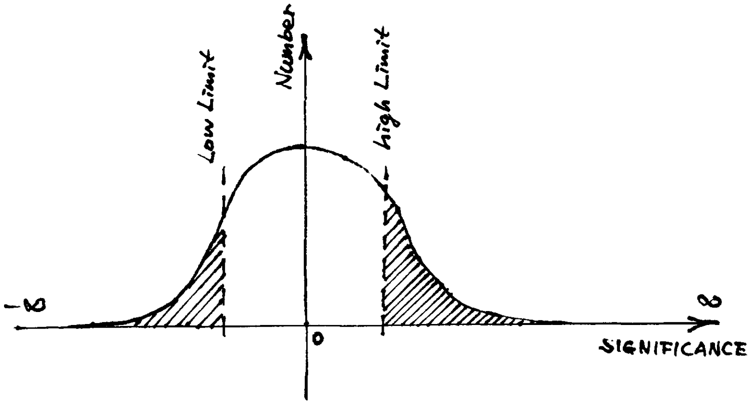




Another improvement is for considering all spatial directions. For simplifying computation, we only measure significances at four directions: up-down, left-right, upper left--lower right, and lower left--upper right. At each of the four spatial directions, a significant degree is measured and theabsolute values are added together to represent the significance of this pixel.

Histogram

After all pixels have been assigned their significance degree, the question of which pixel is more important can be answered. We assume that users will specify how many points are needed (or can be handled by their system). They can give this message by setting a ratio of the number of selection points over the total number of pixels. For selection, a histogram of distribution of significance of pixels is built. See Figure 2. The vertical axis is the number of pixels, while the horizontal axis is the significance value.



Based on this histogram, we now know the status of distribution of pixel significance. A typical distribution curve looks similar to normal distribution curves. More pixels have less significance. The number of points with higher significance values is less. The area under the distribution curve is 1.0 by normal units.

Determination of thresholds

Now, two thresholds can be found from the histogram.

$$\int_{\text{high limit}}^{\infty} \text{Number (significance) d significance} = 0.5 * \text{ratio} * \text{total pixels}$$

$$\int_{-\infty}^{\text{low limit}} \text{Number (significance) d significance} = 0.5 * \text{ratio} * \text{total pixels}$$

Any pixel with a significance value less than the low-limit or greater than the high-limit, should be selected. The sum of two areas beyond the two thresholds low and high-limits (shaded parts) is equal to the ratio the user specified.

Selection of VIP

A simple program scans all pixels and selects those pixels that have significances higher than the high-limit or lower than the low limit.

RESULTS ANALYSIS

There are two test data sets that represent mountainous and flat areas. For each data set, different ratio of VIP points over total pixels are set for selecting VIP points. Then their triangular irregular networks with different accuracies can be constructed, and each new terrain surface is compared with the original terrain surface to see how they match.

CONCLUSIONS

(1) The VIP algorithm can produce better significance distribution at both flat and mountainous areas.

(2) The VIP procedure provides a convenient way to select as many important points as the user needs. For any ratio of needed number of points, a set of points can be selected from all pixels. This set is always relatively more important than other unselected pixels.

(3) The VIP provides a better point set to construct TIN which fits the original terrain surface as faithfully as posible.

REFERENCES

Ballard, D.H. and Brown, C.M., 1982, *Computer Vision,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Fowler, R.J. and Little, J.J., Automatic Extraction of Irregulat Network Digital Terrain Models, *ACM*, 1979 Vol. 4, 199-207.

Guevara, J. A., et al., D.E.M.G.S. A Graphic/Processing System for Quick Inspection and Display of Digital Elevation Models, *Auto-Carto V*, 1982, Virginia.

Peucker, T.K. and Douglas, D.H. Detection of surface-specific points by local parallel processing of discrete terrain elevation data. *Computer Graphics and Image Processing* 4, (1975), 375-387.

Peucker, T.K., Fowler, R.J., Little, J.J. and Mark, D.M. Digital representation of three-dimensional surfaces by triangulated irregular networks (TIN). Technical Report 10, ONR Contract #N00014-75-C-0886, Dept. Geography, Simon Fraser University, Burnaby, B.C., Canada, 1977.

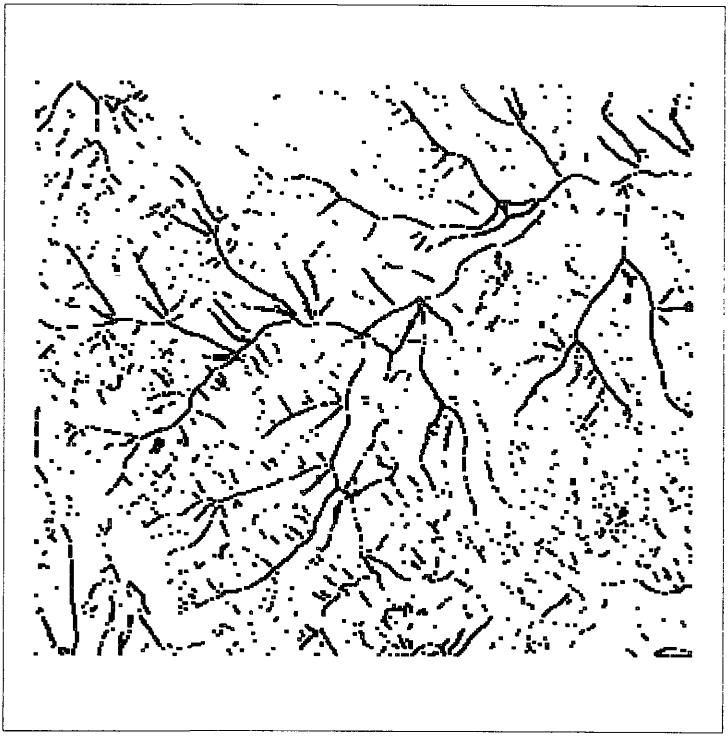


Figure 3. VIP points generated from raster DEM.
(The DEM is original from USGS file. Its size is 351*303.
VIP selected 4591 points. The ratio of VIP point number
over pixel number is 4%.)

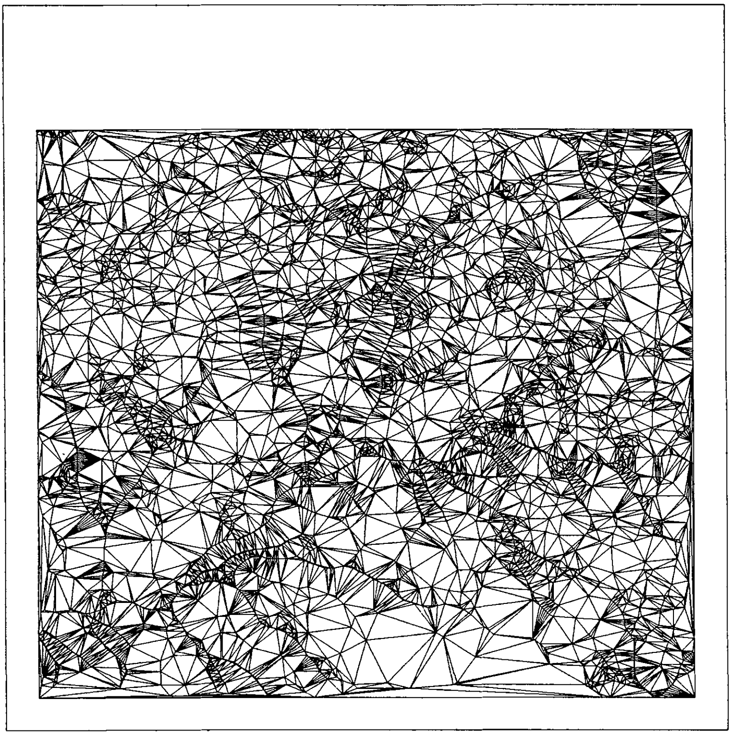


Figure 4. A TIN generated from VIP in Figure 3.
(The TIN has 4591 nodes, 9087 triangles.)

FITTING A TRIANGULATION TO CONTOUR LINES

by Albert H J Christensen

formerly GEONEX Corporation

7836 Muirfield Court
POTOMAC, Maryland 20854
(301) 983 9004

ABSTRACT

This paper presents a technique for creating a triangle mesh that tightly fits a terrain surface represented by a set of digitized contour lines. Basic to the technique is a Medial Axis transformation of a polygon, in this case formed by one or more contour lines. The advantages of using this mesh rather than the well known Delaunay triangulation for computing a gridded Digital Terrain Model (DTM) are discussed, as well as widely used spline interpolation methods. An example illustrates how the Medial Axis relates to the polygon and triangles and thereby facilitates further adjustments to the mesh. More complex adjustments to convert the triangulation into a surface devoid of unnatural features are described. Anomaly-free DTMs can be computed from contours without the supplementary features demanded by interpolation and triangulation procedures in use today. Desk-top computer programs operating on a small area of a scanned contour plate were prepared to test and illustrate the procedures that are outlined.

TWO CONTOUR-TO-GRID METHODS

Converting a given set of contours into a gridded numerical model of elevations, commonly called a Digital Terrain Model (DTM), can be accomplished by two widely different approaches.

Interpolation Method. The better known approach that is called here the 'Interpolation' method consists of the following: Vertical planes passing through each grid point intersect the source contours. Straight lines or planar curves contained in the vertical planes are defined by the intersections, and used to interpolate elevations at the corresponding grid point. There are numerous reports on implementation of this approach and on the nature of the curves used in the process. See references in [8,9].

Triangulation Method. The second, less known method, is called the 'Triangulation' method. The triangulation that constitutes the chief component for converting from contours to grid is performed by an algorithm that selects part or all the points in the source contours and establishes with them a mesh of non-overlapping triangles. From these triangles grid values are computed.

THE DTM IN BETWEEN SOURCE CONTOURS

Before discussing the problems found in interpolated DTMs, the subject of how the DTM is expected to behave in areas devoid of sampling should be examined. Obviously, the replication of source contours from the DTM should be a concern, although by no means the only or the most important one. Until recently, however, and so far as this authour could verify, only those few engaged in the creation or in the inspection of accurate DTMs considered the variations of a DTM away from the sampled areas. A concern so restricted was perhaps due to the lack of a reliable model against which a DTM could be compared.

Recently, two papers [8,9] have been published on evaluations of a number of interpolation techniques. In both papers the source contours are derived from a synthetic surface and converted into gridded DTMs by applying different interpolation methods. The predicted DTM values are then compared to those directly computed from the surface equation. A third paper [14] shows the wide disagreement in areas of low sampling density between derived contours and true contours which were not included in the input data set.

A synthetic surface exhibiting a number of formations similar to those found in topographic surfaces, as in [8], is a very attractive proposition for detecting and measuring DTM undulations. However, for accurate DTMs the maximum deviations allowable are smaller than the 25% of the contour interval which the plots in that paper show as lowest error. The plots in a future article would be perhaps very revealing if the authors would lower the mimimum error to, let's say, 3.5% of the contour interval. This is one of the maximum deviations established for accurate DTMs in flat areas.

DTMs generated to meet such strict specifications have to pass complete and thorough inspections. One of the tests compares a number of grid point values against values sampled from the source document.

Other tests developed for the verification of DTMs are mostly visual. The display of a grid of first and second differences computed from the elevations is an effective test. The differencies tend to highlight areas where the undulations introduced by the splines have propagated in linear or areal patterns, commonly known as 'unnatural' features. Examples are false dams, false depressions and bumps. Also clearly shown are patterns created by the symmetric distribution of intersecting planes, especially strong when only two planes are used.

Not surprisingly, the occurrence of unnatural features is higher where the spatial coherence of adjacent contours'is lower, as in flat areas. Under strict specifications such occurrances must be avoided, which interpolation methods can accomplish if additional linear data is available. Examples are 'fabricated' contours, added to the source contours in flat areas. Other additional lines are used by programs that follow the interpolation of the grid points. Such are the drainage lines, with which the programs perform two functions. First, they introduce breaks in the DTM, and second, they remove any false dams accross the drainage lines. Supplementary drainage and other terrain features are created in low coherence areas, usually in correspondence with strong contour sinuosities, and processed together with the natural drainage lines.

Adding linear features to a contour set is a task that demands a fair amount of training and a good understanding of the entire DTM process. Moreover, it is a manual digitization task and consequently, costly both in labour and in equipment.

THE RULED SURFACE BETWEEN CONTOURS

More interesting than anything in a DTM Quality Control document is the aforementioned selective point verification. A number of elevations at grid points are evaluated, presumably by comparing them with elevations extracted from the source topographic map. How are these elevations computed? Most likely in the same way a topographer of Yesterday interpolated contours. For instance, when he metricized a map. Since the operation was manual, he had to use the simplest procedure that could be carried out with contour lines.

The topographer proceeded according to the assumption traditional in elementary Descriptive Geometry: between contours a terrain surface is ruled and not developable. In other words, along certain straight lines a topographic surface has constant slope. It follows that DTM derived contours ought to be as regularly spaced as possible between source contours or, more formally, that distances between derived contours measured along lines of maximum gradient should be equal.

Needless to say, if a DTM quality control test is modelled on a ruled surface, it makes good sense to design the DTM around the same model.

THE TRIANGULATION APPROACH

The fact that triangulations created from contours have not been implemented as frequently as interpolations may be explained by their degrees of success. This observation does not apply to triangulations of randomly distributed points, such as in meteorology, geology and the like, where triangulations are routinely accepted.

The triangulation of a point set used today for most applications is Delaunay's [11]. Its tendency to yield triangles as well shaped as possible makes it attractive for applications using functions with singularities at very small angles. It is even more attractive because its uniqueness, which in turn makes the task of programming it light. The disadvantages are, first, a special configuration of points that must be considered [11], second, thin, sliver-like triangles along the perimeter of the mesh, which Delaunay algorithms create just to achieve convexity, and third and most important, that if a 'brute force' approach is taken, the processing time may grow beyond realistic possibilities.

Many solutions have been proposed to reduce the growth of the processing time to more manageable limits. Almost all of them exploit the principle of 'Divide and Conquer' lucidly exposed in [1], and ought to be applied to all triangulations of great numbers of scattered data points.

With the exception of those concerned with a distance optimality, all the examples known to this author on triangulations of point sets are Delaunay's. So do the few Contour-to-Grid conversions by triangulation: two commercial Site Engineering packages and the implementations in [10,13.]

Applied to contours, the Delaunay triangulation knowns only of contour points. The fact that the points are connected in the shape of contours is not considered. Consequently, poorly configured triangles may result. A case is that of a triangle edge crossing a contour segment. The triangle edge, now supposedly an element on the terrain surface, may have in correspondance with the contour segment an elevation different to that of the contour. If the crossed contour is higher or lower than both adjacent contours, the error amounts to 100% of the contour interval, see Fig.1. To prevent such configurations all the contour segments should be selected as triangle edges, which is a proposition that invalidates the Delaunay triangulation as applied to the entire set of contour points. See in Figure 1 a catastrophic false dam.

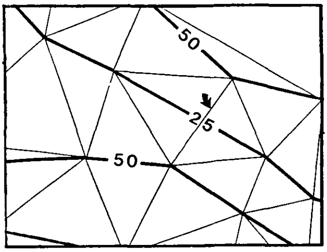


Figure 1. Contour crossing

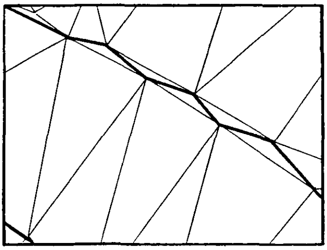


Figure 2. Flattening along contours

A second case is that of a triangle with its three vertices on the same contour. Such triangles cause breaks in the surface and should be avoided if the DTM is to be smooth between contours. Perhaps the most striking result of this poor configuration are horizontal bands of triangles produced along sections of contours, Fig. 2, and beyond the band areas slopes slightly steeper than what they really should be.

Avoiding the crossing of contours. The contour crossing case could be avoided by performing Delaunay triangulations in between adjacent contours, which means that the entire map would be covered with many triangulations, each of them executed independently of the others. Besides avoiding the contour crossing, this procedure will greatly alleviate the processing time problem, since the contours provide natural boundaries for the application of the 'Divide and Conquer' principle. There is no need for artificial divisions when contours are present. Indeed the triangulation of a point set inside a closed shape is not a novel idea, although this author has not yet seen it applied to topographic surfaces. The field of Pattern Recognition offers one example [7]. Heuristic and optimal triangulations, non-Delaunay, of bands limited by successive planar contours, have been proposed in [4,5,6] for the reconstruction of 3D surfaces. However, these references must not be interpreted as suggesting that such techniques can be applied to topographic surfaces. Terrain surfaces are single value functions of two variables, but they can be far more topologically complex than the true 3D surfaces of the type reconstructed using the reported techniques.

Because of the aforementioned topological complexity, the Delaunay triangulations inside a closed shape, with islands added, is not as easy a proposition for computer programming as the general Delaunay triangulation.

Horizontal Triangle Case. The second objection to the general Delaunay triangulation, triangles with their three vertices on the same contour, is not so easily removed. It will be discussed later.

It must be noted that the critiques in this paper to the general Delaunay triangulation of contour maps ends precisely with the triangulation. After the triangulation has been established, other procedures may be used to reshape it, for instance, exchanging edges so that they would not intersect contours, while others might even fit high order surfaces to the planar triangles for computing grid points. The availability of such follow-up procedures does not negate the conclusions of this paper.

THE MEDIAL AXIS TRANSFORMATION

This author's opportunity for experimenting with some old ideas on how to create a ruled surface from contours arose from the need for a procedure to thicken or widen line features. This need was satisfied by developing a 'Parallel Pairs' procedure, published elsewhere [3], that also suggested possibilites for solving some other problems. One of these problems was the 'Medial Axis Transformation', an operation which turned out to be basic to the Contour-to-Grid solution described in the next sections.

The Medial Axis [7,12] or midline, of a closed shape or polygon, is, rougly, a network of lines whose elements are equidistant from the closest pairs of elements in the shape. The Medial Axis is defined in vector environments. In a raster environment, it corresponds to the skeleton of a shape. In computer operations, the Medial Axis transformation corresponds to the raster 'thinning' or 'skeletonizing' operation with which commercial scanners are often provided. The thinning operation, coupled with the raster-to-vector conversion that comes with commercial scanners, make a fast and robust tool for generating the Medial Axis. There is an abundant literature on thinning, see for instance [2]. The vector mode operation seems to be less popular.

The output of the Parallel Pairs procedure is a set of polygons nested inside the input polygon, see Fig.3, with which the determination of the Medial Axis is accomplished in an efficient way. Exhaustive searches become unnecessary, as reported in [3], because the Parallel Pairs are loaded with pointers that indicate through which points the Medial Axis should be threaded. Pointers extracted from the parallel pairs are loaded into the Medial Axis as well, linking its elements to the equidistant edges of the input polygon.

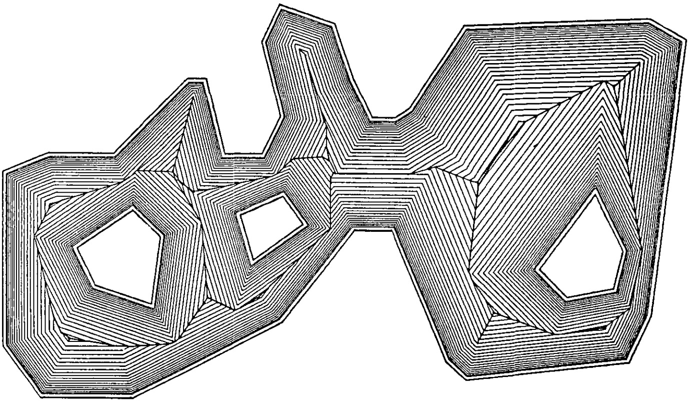


Figure 3. Dense parallel pairs and Medial Axis.

Advantages of determining the Medial Axis in raster rather than in vector mode are simplicity in programming, robustnéss, and perhaps time performance. Disadvantages are inflexibility, lack of structure and the need for a raster-to-vector conversion that follows the thinning operation. Inflexibility arises from the fixed resolution of a raster system. All the polygons in the file, irrespective of their particular shapes and dimensions, are processed with the same resolution.

Lack of structure refers to the absence of pointers and other features that facilitate further operations. The raster skeleton cannot be related to the polygon edges, at least in today's commercial software. Nor is easy to see how it could possibly be done, when the input is just a raster image. If the skeleton and the polygon together must be processed further, as in the case discussed here, the lack of structure would surely offset any time saved by the raster mode operation.

On the contrary, the vector approach offers flexibility and a potential for structure. Its flexibility is found in the wider range of the arithmetic that is used. The Parallel Pairs procedure, as part of a Medial Axis Transformation, increases that flexibility by providing the very significant option for changing offsets in the nesting of polygons. See Fig. 3. It also provides structure in the pointers referred to earlier.

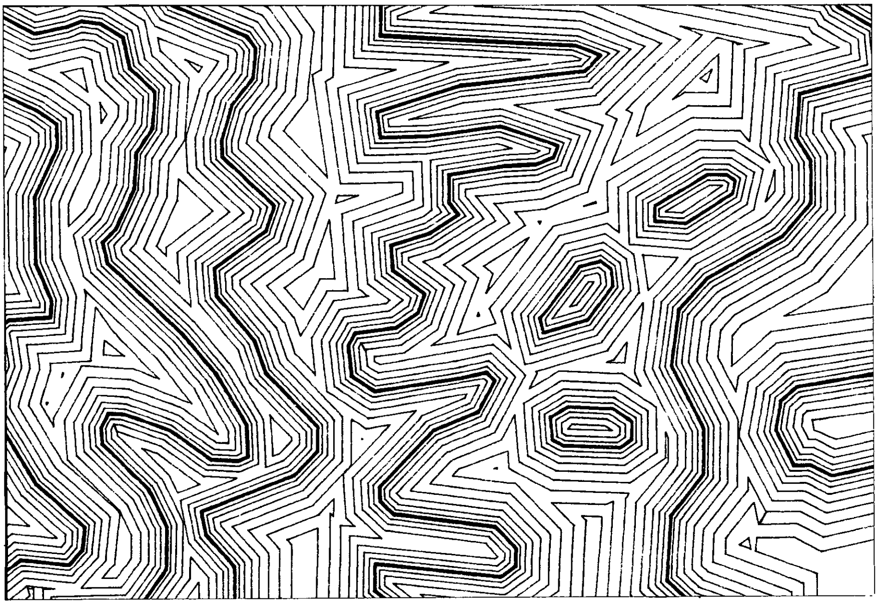


Figure 4. Parallel pairs in a section of a contour sheet

Figure 4 shows a small section of a contour sheet, with nested polygons generated with offsets greater than those used for figure 3. The original was a 1:50000 topographic sheet, scanned at 16 lines/mm resolution. At the original scale the area illustrated here measured

2 by 1.2 cm. The nested polygons were created in the areas bound by adjacent contours, in a process that was run separately for each area. They were then merged and plotted. The smallness and low speed of the desk-top computer used to prepare the software, resulted in data sets that are very limited in complexity.

THE TRIANGULATION OF A POLYGON AND ITS MEDIAL AXIS

As noted earlier, a Delaunay Triangulation, be it applied to disconnected contour points or executed inside a closed contour polygon, in many cases will select the three vertices of a triangle from the same contour, and that these horizontal triangles introduce breaks that do not provide a natural gradient to the surface. This is a problem that could not be ignored.

That problem can be solved by using the Medial Axis because there are always points on the Axis that can be connected to the contour points. The triangulation uses the Medial Axis points to bridge the spans between contours. Furthermore, because it is executed between contours, this triangulation does not cross them. Because each Axis can be given the mean of the contour elevations, the triangles on both sides of the Axis will have the same slope. With this procedure the Medial Axis itself will not turn out to be an unnatural feature.

In the program prepared to test the proposed solution, the vertices of the triangles are selected with a simple rule: the base of a triangle is defined by two consecutive points, either from the contours or from the Axis. If from the contours, then the apex of the triangle is selected from the Axis, and vice-versa. The pointers in the Axis tell the process from which entity, contour or Axis, to select the next base. Executed in this way, the triangulation program is extremely fast. Fig. 5 shows the triangles established in the same small contour shape of Fig.3.

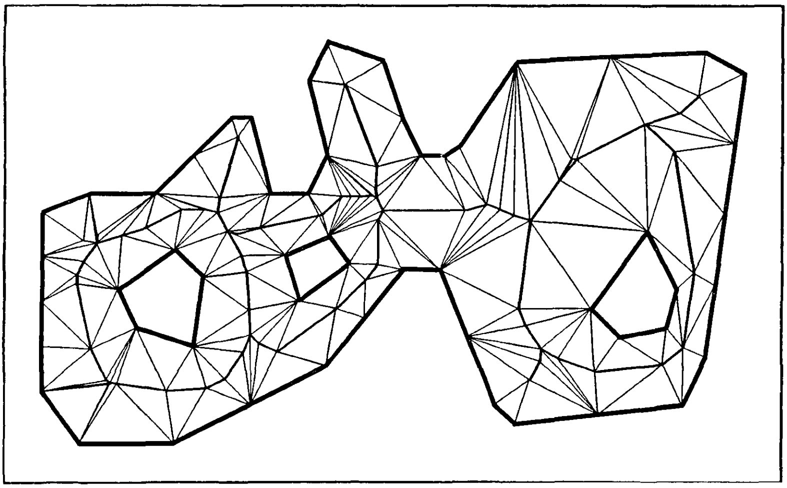


Figure 5. Triangulation of contour shape and Medial Axis

Assigning elevations to the Medial Axis. A quick look at a few shapes and their Medial Axes leads to the following classification. As regards points, only the endpoints of a line, called nodes, are considered. The number of lines incident to a node is called the 'degree of incidence'. These are Graph Theory terms. Lines can be classified as open or closed. A closed line has only one node and this is of degree 2. As Medial Axis, a closed line is a rarity. Open lines are classified here as Main lines and Branches. A Main line has its two node of degrees 2 or higher, or both of degree 1. Branches have one and only one node of degree 1, the dangling endpoint. With this classification is is possible to conclude, in a general way, that Main lines are connected, by means of the triangle edges, to two different contours. Branches to only one contour. See Fig. 5.

One part of the triangle vertices, those on the contours, can only be given the corresponding contour elevation. As for assigning elevations to the rest, on the Medial Axis, it is necessary first to make an assumption on which elevation to give to the Medial Axis.

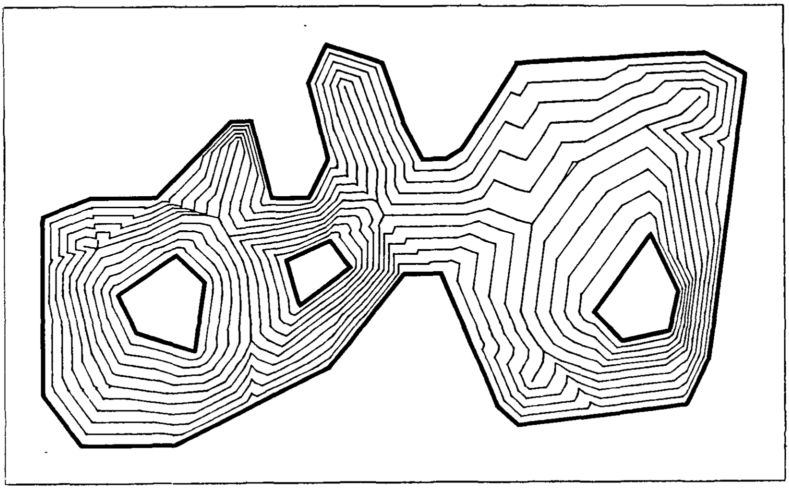


Figure 6. Derived contours for a Medial Axis with constant elevation

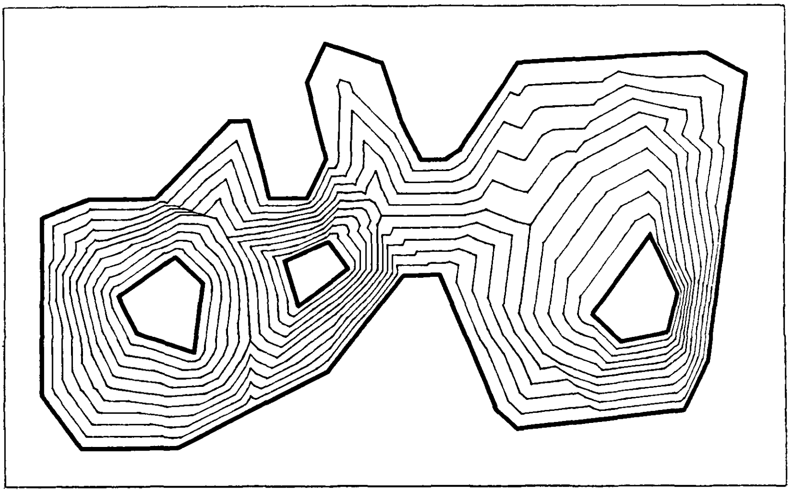


Figure 7. Derived contours for a Medial Axis with adjusted elevations

To avoid turning the Main lines into unnatural features, their points should be given the mean of the elevations of the contours with which each Main line is connected. The same cannot be done with the Branches. If they are given the same mean as the Main lines, the result will be a strong gradient located at the end of the Branch. Fig. 6 shows the contouring of the triangulation executed under this assumption. Notice how the contours are crowded at the end of the various Branches. A more natural look and a better approximation to a ruled surface is achieved by assigning variable elevations to the Branches. Fig. 7 was obtained with an option of the experimental software, which assigns to the Branches' points elevations proportional to the distances measured along the Branch from the non-dangling node. The proportionality is established between the difference in elevation, mean minus contour, and the length of the Branch plus the length of the shortest edge that connect the dangling end of the Branch with the contour. The better quality of the result, compared with the one obtained by giving constant elevation to the Axis, is evidenced in Fig.7: better spaced derived contours along all the Branches.

THE CONTOUR-TO-GRID PROPOSED SOLUTION

The Medial Axis transformation and the simple triangulation that comes after it are just two, if important, steps in the proposed solution. To make the operation of the Parallel Pairs possible, and in general, to improve the time performance of the software, as well to simplify its overall design, the input contours must be preprocessed.

Preprocessing of contours. First, all the contours are assumed distinct and without gaps. Those that reach the map borders must be turned into closed lines. In doing so, the closing lines ought to be such that the nested polygons would have the proper orientation when crossing the map borders. Figure 8, to be inspected together with Figure 4, shows how this was done with a simple program.

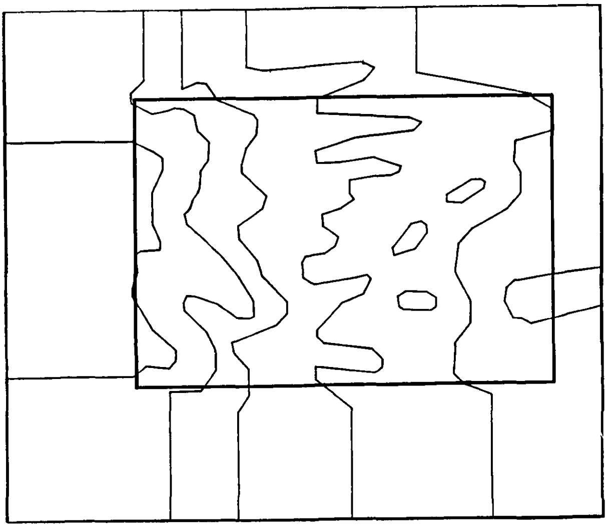


Figure 8. Contours closed beyond map borders

Second, adjacency relationships and containment should be introduced in the source contours and spot heights. These relationships are needed, inter-alia, for assembling pairs of contours into the shapes to be triangulated. The abundant literature on this subject make unnecessary any explanations. Third.Although not strictly needed, it is convenient to obtain from the contours some measures of size and their proximity to each other.

The last two requirements can be best satisfied by programs operating in raster mode at the time the contours are vectorized. If a commercial vectorizer is being used, it will be necessary to rasterize the contours before running these programs.

After the preprocessing of the contours, the Medial Axis is determined, as already described, for each of the areas enclosed by one or by two succesive contours. This step is followed by the triangulation of those areas, which in turn is followed by the computation of the grid.

The Grid from Triangles. The computation of the grid values from the triangles, if these are considered planar, is a simple operation and does not merit any reference here.

However, if a smoother surface is desired, the triangles may be turned into curved patches that preserve continuity accross edges. There are many ways of defining such patches. The issue, in the view of this author, is not how to do it, but whether or not to do it, and the answer, on technical grounds only, is no. Yet, if some smoothing is still wanted, it will be enough to break the triangles at their half heights, and to assign to the breaks elevations that reflect a curvature along lines of maximum gradient. Of course, in directions normal to these lines, any smoothing would be still more superfluous.

Tops and Depressions. Shown in a contour sheet as empty closed lines, they have preocupied the advocates of triangulations since very early. In most cases, these closed contours do not include spot heights in sufficient numbers and proper distribution to ensure a good reconstruction of the terrain. The results are 'truncated tops' and 'flattened depressions' in the DTM. To produce a correct DTM the user will have to create automatically, or by hand, the right number of spot heights in the right places. The Medial Axis provides an automated solution to this problem. Figure 9 shows the Medial Axis and the resulting triangulation of a top contour.

Assigning elevations to a top contour or to a depression is done either by resorting to the spot heights or to the triangles adjacent to the contour in question. From those triangles slopes can be extracted and then applied to the triangles inside the top contour Or depression. The procedure followed for assigning elevations is very much the same used for contour polygons in the general case.

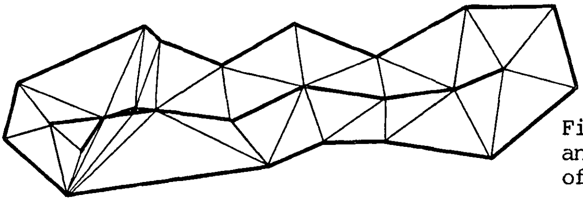


Figure 9. Medial Axis and triangulation of top contour

CONCLUSIONS

The Medial Axis provides the means for triangulating contours in optimal configurations, from which an accurate gridded DTM can easily be computed. This DTM behaves like a ruled surface, and consequently, does not exhibit any of the unnatural features introduced by spline interpolations, nor the breaks and false dams created by Delaunay triangulations. The discussions of these techniques were done with strictly specified DTMs in mind. It is hoped, however, that the precise fit of the triangulation described here will facilitate the introduction of accurate DTMs into fields where profiling and cross-sectioning are still prevalent.

AKNOWLEDGEMENTS

This paper would never have been produced had not the author been provided the encouragement and support of his former employeer, GEONEX Corporation.

REFERENCES

[1] Bentley, J L and Shamos, M I 1978, Divide and Conquer for Linear Expected Time: Information Processinq Letters, Vol.7,pp. 87-91

[2] Bookstein, F L 1979, The Line-Skeleton: Computer Graphics and Imaqe Processinq, Vol. 11, pp. 123-137

[3] Christensen, A H J, Parallel Pairs in Automated Cartography: to be published in Cartoqraphica, Vol. 24, No. 1

[4] Fuchs, H, Kedem, Z M and Uselton, S P 1977, Optimal Surface Reconstruction from Planar Contours: Comm.of the ACM, Vol.20, No.10

[5] Gannapathy, S and Dennehy, T G 1982, A new General Triangulation Method for Planar Contours: Computer Graphics, Vol.16, No.3, pp.69-74

[6] Keppel, E, 1976, Approximating Complex Surfaces by Triangulation of Contour Lines: IBM Journal of Research and Development, XIX

[7] Lee, D T 1982, Medial Axis Transformation of a Planar Shape:IEEE Transac. on Patt.Analysis and Machine Intell., PAM14,4, pp.363-368

[8] Legates D R, and Willmott, C J 1985, Interpolation of Point Values from Isoline Maps: The American Cartoqrapher, Vol. 13, No.4, pp. 308-323

[9] Loon, J C, and Patias, P G 1985, Digital Terrain Elevation Model Analysis, Report ETL - 0393, U.S. Army Corps of Engineers.

[10] McCullagh, M J 1983, Transformation of Contour Strings to a Rectangular Grid Based Digital Elevation Model: Proc. Euro-Carto II.

[11] McLain, D H, Two Dimensional Interpolation from Random Data: The Computer Journal, Vol. 19, No. 2, pp. 178-181

[12] Shapiro, B, Pisa, J and Skansky, J 1981, Skeleton Generation from x,y Boundary Sequences: Comp. Graph.and Imaqe Proc. Vol.15, pp.136-153

[13] Witzgall, C, Bernal, J and Mandel, B, 1986, On Sampling and Triangulating Large Digitized Contour Data Sets. National Bureau of Standards and U.S. Army Engineer Topographic Laboratories

[14] Yoeli, P. 1986, Computer Executed Production of a Regular grid of Height Points from Digital Contours, The American Cartoqrapher, Vol.13 No.3, pp. 219-229

MEASURING THE DIMENSION OF SURFACES: A REVIEW AND APPRAISAL OF DIFFERENT METHODS

André G. Roy
Ginette Gravel
and Céline Gauthier
Département de Géographie
Université de Montréal
C.P. 6128 Succ. "A"
Montréal Qc
Canada H3C 3J7

ABSTRACT

The concept of fractals is being widely used in several cartographic procedures such as line enhancement, surface generation, generalisation, interpolation and error estimation. Such applications rely on the estimation of the fractional dimension (D) of lines and surfaces. Measuring D for surfaces can be achieved from contours and profiles extracted from the surface or from the variability of the surface taken as a whole. In a fractal and self-similar terrain, the values of D should be in agreement regardless of the method used. Mark and Aronson (1984) applied the variogram technique to DEM and observed sharp changes in D with scale suggesting that terrains are composed of nested structures with a highly disorganised and complex component ($D=2.6$) in the long range and a smooth component ($D=2.2$) in the short range. The high dimensions may not reflect the terrain itself but be the result of combining residual anisotropic effects at long distances. Tests performed on DEM (or portions of DEM) show that the short range dimensions of the surface variogram are consistent with those extracted from profiles and contours ($2.0 < D < 2.3$). Systematic variations of D with altitude and location were also observed indicating a lack of self-similarity in spite of the apparent self-similarity of the surface variogram.

INTRODUCTION

Several applications of fractals to cartographic lines and surfaces are now well entrenched in the literature. The fractional dimension which characterizes the geometry of a line or surface is a powerful tool for analysis, description and generation of cartographic data. Cartographic lines appear to be more easily reduced to fractal analysis than surfaces. One problem that arises when dealing with surfaces is the difficulty of estimating their fractional dimension. This difficulty is partly explained by the availability of several methods for the computation of the dimension and by the lack of self-similarity in natural terrains. This paper addresses these problems and discusses some implications for cartography.

FRACTALS AND THEIR APPLICATION IN CARTOGRAPHY

This brief review of fractals will emphasize the properties of fractal sets and their applications in the field of cartography. Fractals were introduced by Mandelbrot (1977) to describe, among other things, irregular lines and surfaces. Strictly speaking, fractal applies to entities which have an Hausdorff-Besicovitch dimension (D) greater than the topological dimension. The value of D characterizes the intricacy or the jaggedness of the entity. Lines will have dimensions varying from 1 to 2 while surfaces are described by values of D ranging from 2 to 3. As D increases towards the upper value of the range, the entity becomes highly complex and intricate and the process associated with the line (or surface) is space-filling.

Fractal models of lines and surfaces may be created through fractional Brownian processes (Mandelbrot 1975; Goodchild 1982; Burrough 1983). In practice, fractional Brownian functions may be generated from fractional Gaussian noise. Several properties of such processes are noteworthy. First, the variogram of fractional Brownian functions is described by

$$E[(z_i - z_{i+h})^2] = h^{2H} \qquad (1)$$

where h is the distance (or lag) between two points and z_i, z_{i+h} are the values observed at point i and i+h respectively. The variogram takes on the form of a power function in which H should vary between 0 and 1. In the case of profiles

$$D = 2 - H \qquad (2)$$

while for fractional Brownian surfaces

$$D = 3 - H. \qquad (3)$$

Secondly, the covariance function of such processes also display a relationship with H and consequently with D. As H increases toward its upper limit, the positive auto-correlation between neighbouring values is very strong and the realization of the process is very smooth. The process is Brownian for H=0.5. When H gets below 0.5, then the process tends to become anti-persistent and negatively auto-correlated. Profiles and surfaces generated with values of H lower than 0.5 are very jagged and erratic (Burrough 1983; Goodchild 1982; Culling 1986) and will display a rapid succession of peaks and throughs. Similar interpretations of D in geostatistical terms may be found through the applications of the power spectrum (Mandelbrot 1982; Pentland 1983).

Fractal models display the property of self-similarity which may be viewed strictly as a cascading mechanism of a fractal generator (Mandelbrot 1977). Self-similarity also implies that H and therefore D, the fractional dimension of an entity, is constant with changes in scale. Thus, small portions of the process are replicates of the global structure. By looking at the realization of a fractional Brownian process, one cannot infer its scale. For fractal surfaces, the same process has operated across the whole entity and self-similarity is associated with isotropy, that is the lack of directional bias in the geostatistical properties of the surface. Thus, profiles extracted from the fractal surface will have the same dimension than that of the surface itself less one. Contours and coastlines will also display the same dimension than the profiles. Self-similarity implies dimensional consistency among the lines and the surface.

Because it deals with the effect of scale on the metric of lines and surfaces, the concept of fractals has proven to be very useful to cartographers. The addition (or elimination) of details into a cartographic entity say a line is a process that may be consistent with the fractal geometry (Buttenfield 1985). Line degeneralization pionnered by Dutton (1981) used the fractal dimension to introduce details into a generalized line. This fractalization process enhances the line. Algorithms for fractal interpolation are well known (Fournier et al. 1982) and are also used to generate terrain profiles (Frederiksen et al. 1985). Muller (in press a) proposed to rely on the property of self-similarity as a standard to assess the quality of line generalization. A generalized line should be self-similar to the original. Furthermore, efficient line generalization may be achieved through the application of walking-step algorithm which is a straightforward application of fractals (Dubuc 1985; Muller in press b).

Errors in sampling and measuring from cartographic data are also related to fractals as it was shown by Goodchild (1980). Because errors increase with the complexity of the entity, they will increase with D. Moreover, fractal dimensions may be used to determine the sampling density required to capture the variability of the phenomenon. Blais et al. (1986) and Dubuc (1985) provided methods of specifying the optimal resolution which are based upon the fractal behavior of the lines or profiles.

Surfaces and more specifically terrain generation has relied heavily on fractals (Mandelbrot 1975; Fournier et al. 1982; Goodchild 1982). Although fractional Brownian landscapes with dimensions of 2.2-2.3 achieve realistic representations of the surface of the earth, little is known about the dimensionality of natural terrains. The estimation of the dimension of a natural surface may be problematic, however. Different methods are available for computing D, all of which should yield similar results if the assumption of self-similarity holds. Given that the processes acting upon the landscape vary with scale, self-similarity may not exist at all scales and for all natural terrains (Mark and Aronson 1984). Goodchild (1982) reported systematic variation in D as we climb from the shorelines to the summit of Random Island. Mark and Aronson (1984) suggested that many landscapes are generated by nesting structures of varying complexity. The problem of estimating D for a surface is compounded by the fact that the different methods may be applied to data coming from different sources which may or may not incorporate a high degree of cartographic generalization.

MEASURE OF THE FRACTAL DIMENSION OF SURFACES

The properties associated with fractional Brownian processes are used to estimate the values of D which may be computed from the variograms of the surface (eq. 1 and 3) or of the profiles (eq. 1 and 2). These dimensions should be consistent with those extracted from contours or coastlines. Mark and Aronson (1984) presented a method to construct the surface variogram of a Digital Elevation Model (DEM) recorded along a regular grid. They proceeded as follows. They randomly selected 32000 pairs of points. Each point had to be within the largest circle drawn within the map. For each pair of points, distance and the squared difference in elevation were computed. The set of measurements was then divided into 100 distance classes of equal size and then the variance of each class was computed. Classes with less than 64 observations were omitted.

They applied their method to seventeen 7½ quadrangles obtained from the USGS. In 15 out 17 cases, they reported that the surface variogram could be described by at least two markedly different slopes (H) and thus two dimensions. At short ranges, for distances smaller than 0.6 - 1 km, they observed relatively low values of D ($D < 2.48$ and close to 2.1) while for longer distances (1 to 4 km) they noted a sharp increase in D sometimes up to 2.8. The average D for this range is 2.6 thus suggesting a very irregular terrain. The low values of D in the short range are more consistent with what has been previously reported in the literature (see Culling 1986) and identifies the strong positive auto-correlation at the hillslope scale. The higher D values are more problematic, however, and although the authors suggest a structural interpretation of the high irregularity, this result is unexpected.

Several problems seem embedded in the method presented by Mark and Aronson (1984). First, the sampling plan is biased towards the long range of the variogram and the random selection of pairs of points within a circle will always generate many more middle and long distances than short ones. Thus, by allowing the random selection of pairs of points, the emphasis is put on the part of the variogram which is farther away from the origin. In view of the fact that the analysis of the variogram tends to rely on the proximal part of the plot, this sampling bias may be important and yield unreliable D values in the short range. Furthermore, normal use of variograms tends to exclude the variances computed for the range of distances farther away than ¼ of the maximum distance on the map. Finally, the surface variogram may be viewed as a composite of profile variograms which may display different characteristics according to the direction. Such directional biases will represent anisotropies of the terrain. In order to avoid a sampling bias, variances should be computed for selected distance classes. A point could be randomly chosen within the largest circle contained in the map. The selected distance could be walked from that point in a direction which would be determined randomly. If the end point of the walk is outside the circle, the pair of points would be rejected from the analysis. This scheme is advantageous because we control the distance classes which could be specified in a geometric progression and also because the number of pairs in each class could be determined a priori.

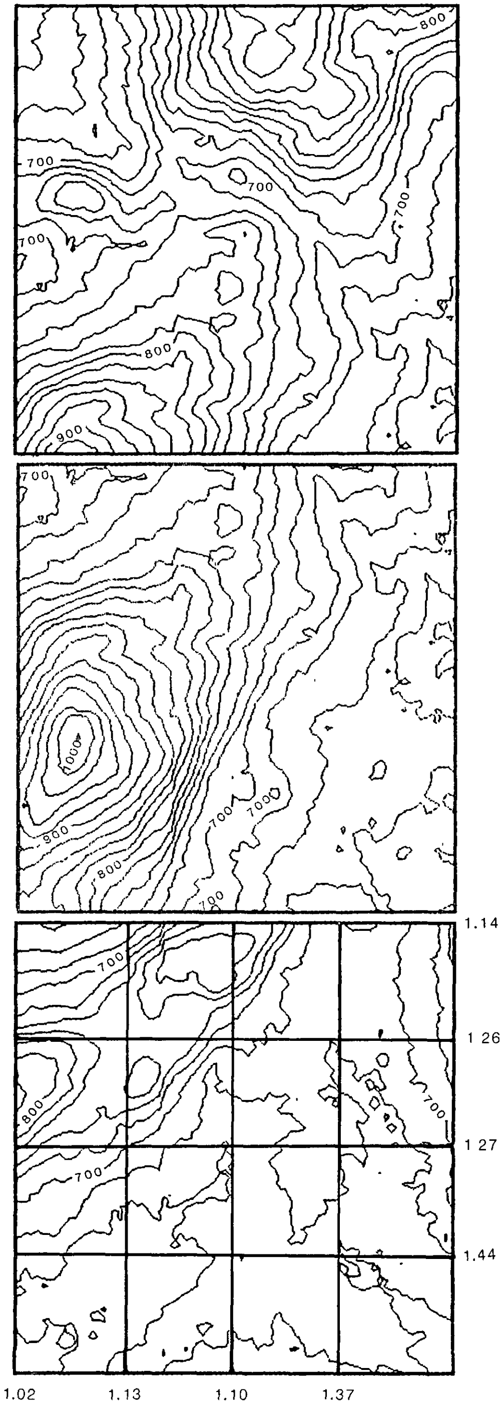
Nonetheless, the surface variogram should be preceded by an analysis of profile variograms which would allow to detect the presence of anisotropies. The search for directional bias can only be done in the NW, NE-SW, NW-SE, NS directions, however. All other directions would involve interpolated values and the variogram would not reflect the variability of the raw data. Profile variograms are simply build from the systematic sampling of all possible pairs of points separated by a distance h. Thus, the estimates of the variance at a longer range are derived from fewer pairs of points and the variogram should be reliable for distances shorter than one fourth of the maximal distance.

The dimension of a surface may be found from the dimensions of the contours and coastlines. In doing so, we are concerned with three problems. First, several techniques are currently used to estimate D for such lines. Most of these techniques involve the estimation of the rate of change in length with an increase in the sampling interval. The slope (b) of the log-linear relationship between the length of the line and the length of the divider used to measure it is given by

$$b = 1 - D . \tag{4}$$

Other methods rely on cell counting algorithms (Goodchild 1982; Shelberg _et al_. 1983). Goodchild (1982) compared several techniques and obtained higher estimates of D when length was measured from a cell counting method. Reliability of each method is difficult to assess, however. Secondly, the selection of the contours that we submit to fractal analysis may be critical. In a self-similar terrain, this would not be of concern since all contours display similar complexity. Natural terrains may exhibit systematic variations in complexity as was pointed out by Goodchild (1982). Shelberg _et al_. (1983) suggested that a set of contours should be _used_ in the analysis. Finally, should the contours be taken from the maps (Goodchild 1982) or derived from the DEM itself (Shelberg _et al_. 1983) ? If cartographic generalization preserves self-similarity, then the source of the data would not affect the estimation of D. Such a postulate remains to be shown.

FIGURE 1: Three 80 x 80 windows extracted from the DEM



A - Fluvial

B - Summit

C - Glacial

APPLICATIONS TO DIGITAL ELEVATION MODELS

The different methods of calculating D were applied to a USGS DEM of an area located in the White Mountains at the border of Quebec, Maine and New Hamshire (Moose Bog 7½ Quadrangle). Maximum relief in the quadrangle is 700 m. Three 80 x 80 windows illustrating different landscapes within the area - a fluvial landscape at the headwater of a stream (Fig. 1 A), a summit area (Fig. 1 B) and a valley filled with glacial sediments (Fig. 1 C) - were also submitted to fractal analysis. For the whole quadrangle and the three windows, D was estimated using four techniques:

- the surface variogram sampled using the fixed length technique;
- the variograms of profiles taken across the DEM in the EW, NS directions and along the diagonals;
- the contours digitized from the topographic map;
- the contours threaded into the altitude matrix.

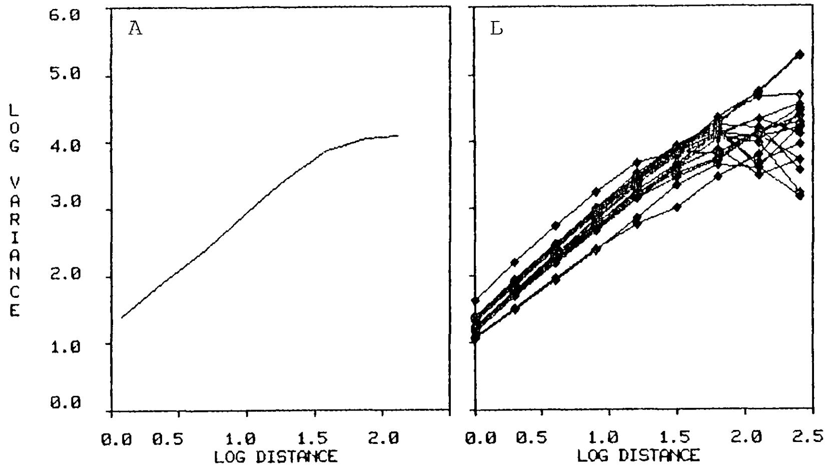
The dimensions of contours were evaluated using the dividers technique.

The surface variogram (Fig 2 A) obtained from the whole DEM shows an initial straight segment up to a lag of 2.0 km (64 pixels) with a constant slope (H = .84). D is therefore equal to 2.16 and it indicates a strong positive auto-correlation of elevations. The distal part of the variogram for longer lags also has a trend (H = .18). The break in slope is sharp as was the case of the examples presented by Mark and Aronson (1984) but it occurs close to the limit of reliability of the variogram (one fourth of the maximum distance is 79 pixels). The variances computed for the surface result from the composite effects of the profiles. This is shown in Figure 2 B where all 20 profile variograms are plotted. We note that the slopes of the initial segment are relatively constant while the distal parts of the variograms are highly variable. The residual trend observed in the surface variogram is clearly the amalgam of highly variable behaviors at longer distances and cannot be meaningfully interpreted. Thus, we conclude from the surface variogram that it describes an apparently self-similar terrain. This conclusion was also confirmed by plotting the profiles to scale and sampling them at various intervals. Smoothness of the terrain was always evident.

TABLE 1: Dimensions computed from different methods for the DEM as a whole

METHOD	D	DMIN	DMAX
Surface Variogram	2.16	----	----
EW Profile Variograms(9)	1.13	1.06	1.19
NS Profile Variograms(9)	1.17	1.09	1.28
Diagonal Profile Variograms(2)	1.21	1.17	1.25
Digitized Contours(13)	1.17	1.06	1.33
Threaded Contours(47)	1.09	1.01	1.28

FIGURE 2: Surface (A) and profile (B) variograms for the whole DEM



The average values of D computed from all methods are strickingly consistent (Table 1). The low D confirm the smoothness of the landscape despite a great amount of vertical relief in the area. Minimum and maximum D values show some variability in the estimation of D. The variability is greater for the digitized contours. This is explained by the sampling plan which attempted to capture the whole range of contour complexity.

TABLE 2: Dimensions computed from different methods for three 80 x 80 windows

METHOD	WINDOW 1	WINDOW 2	WINDOW 3
Surface Variograms	2.13	2.10	2.21
EW Profile Variograms	1.11	1.10	1.28
NS Profile Variograms	1.17	1.13	1.15
Threaded Contours	1.07	1.08	1.10

The comparison of D values obtained for the three windows shows differences among the terrain complexities especially when we look at the values obtained from the variograms. The valley filled with glacial deposits (Fig. 1 C) has a higher complexity than the summit area (Fig. 1 B) or the fluvial landscape (Fig. 1 A). This difference had to be expected and becomes even more important when the dimensions of individual profiles are compared (Fig. 1 C). All profiles that entirely cut through the valley bottom have a high dimension ($D \simeq 1.37 - 1.44$) while those on the hill side are very smooth ($D < 1.10$). Some profiles combine the attributes of both types of terrain. Thus within a relatively small terrain we assist to rapid changes in complexity depending on the nature of the sediments. At this scale, the lack of self-similarity is shown through a juxtaposition of terrain rather than by nesting smooth within complex structures as is evident from an examination of the contours. Contours become less intricate with altitude (Fig. 3). This is due to the erratic nature of the

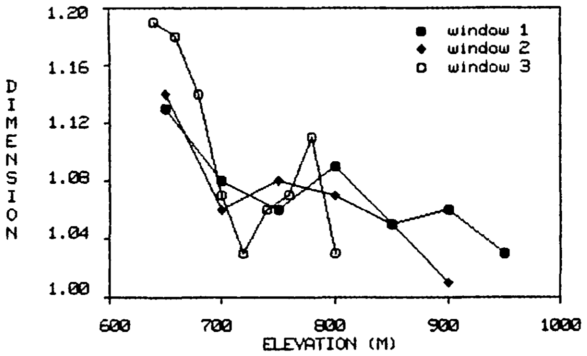


FIGURE 3: Variation of D with elevation

glacial deposits which were laid upon the valley bottom and to the gradual disappearance, as we climb towards the summits, of the crenulations associated with fluvial erosion.

DISCUSSION

Despite the apparent self-similarity of the whole quadrangle, spatial variations in D occur within the DEM. These effects are not detected from the surface variogram because they are not scale-related and the averaging process cancel their individual effect. Goodchild (1982) has also reported similar changes in the fractional dimensions of contours with altitude. In fact, one should anticipate that the dimensionality of most natural terrains should vary spatially. Variations in processes and/or structures may be responsible for these changes in dimensions. For example, creep will produce smoother surfaces than rill erosion. Systematic variations in the dimension within the surface bear important cartographic consequences. For instance, the degeneralization of contours should not be carried out using a unique fractalization process. Elevation and physiographic location must be used to guide the interpolation and enhancement procedures. A similar rationale also applies to terrain sampling as the optimal density should be a function of terrain complexity. Hilltops (in this case study) should be represented with fewer points than the valley floors. The fractal description of a surface should provide useful information for terrain generation and reconstruction and more attention should be given to the fractal signature of characteristic terrains (e.g. fluvial, morainic, eolian landscapes). This conclusion is not unlike that of Mark and Aronson (1984) who viewed the nested structure of terrains as a key component of surface generation. We suggest, however, that the nesting effect is not terrain-related at least in the range of distances where it was observed but rather that the detection of self-similar patches of terrains could be used advantageously by cartographers.

ACKNOWLEDGEMENTS

We acknowledge the financial support of EMR, Canada.

REFERENCES

BLAIS, J.A.R., CHAPMAN, M.A. and LAM, W.K., 1986, Optimal interval sampling in theory and practice: Proc. 2nd Conf. Spatial Data Handling, 185-192.

BURROUGH, P.A., 1983, Multiscale sources of spatial variation in soil: I. The application of fractal concepts to nested levels of soil variation: Jour. of Soil Sci., 34, 577-598.

BUTTENFIELD, B., 1985, Treatment of the cartographic line: Cartographica, 22, 2, 1-26.

CULLING, W.H.E., 1986, Highly erratic spatial variability of soil-pH on Iping Common, West Sussex: Catena, 13, 81-98.

DUBUC, O., 1985, La résolution optimale dans la généralisation de lignes, Unpublished M.A. thesis, Dep. de Géographie, Université de Montréal.

DUTTON, G.H., 1981, Fractal enhancement of cartographic line detail: The American Cartographer, 8, 23-40.

FOURNIER, A., FUSSELL, D. and CARPENTER, L., 1982, Computer rendering of stochastic models: Graphics and Image Processing, communication of the ACM, 25, 371-384.

FREDERIKSEN, P., JACOBI, O. and KUBIK, K., 1985, A review of current trends in terrain modelling: ITC Jour., 101-106.

GOODCHILD, M.F., 1982, The fractional Brownian process as a terrain simulation model: Modelling and Simulation, 13, 1133-1137.

GOODCHILD, M.F., 1980, Fractals and the accuracy of geographical measures: Math. Geo., 12, 2, 85-98.

MANDELBROT, B., 1975, Stochastic models for the earth's relief, the shape and the fractal dimension of the coastlines, and the number-area rule for islands: Proc. Nat. Acad. Sci. USA, 72, 3825-3828.

MANDELBROT, B., 1977, Fractals, Form, Chance and Dimension, San Francisco, Freeman, 365 p.

MANDELBROT, B., 1982, The Fractal Geometry of Nature, San Francisco, Freeman, 468 p.

MARK, D.M. and ARONSON, P.B., 1984, Scale-dependent fractal dimensions of topographic surfaces: An empirical investigation, with applications in geomorphology and computer mapping: Math. Geo., 16, 671-683.

MULLER, J.-C., in press a, Fractal dimension and inconsistencies in cartographic line representations: The Cartographic Journal.

MULLER, J.-C., in press b, Fractal and automated line generalization: The Cartographic Journal.

PENTLAND, A.P., 1983, Fractal-based description: Proc. of I.J.C.A.L., 973-981.

SHELBERG, M.C., MOELLERING, H. and LAM, N., 1983, Measuring the fractal dimensions of surfaces: Proceedings, AUTO-CARTO 6, 319-328.

Stability of Map Topology and Robustness of Map Geometry

Alan Saalfeld
Statistical Research Division
Bureau of the Census
Washington, DC 20233
(301) 763-4506

ABSTRACT

A map's topology and its geometry are grounded in the mathematical theory of cellular structures on continuous surfaces. We say that the map's geometry, the actual physical positioning of the features on a surface, is a realization of the map's topology, which refers to the relative positioning of the features. A single topology has many geometric realizations, any two of which are related by some "rubber-sheeting transformation." Any computerized implementation of the geometry of a map, however, requires a discrete approximation of the point data of the map and a rounding of geometric positioning of those points. The implicit or explicit rounding required will re-position the points, which may in turn change the topology and give rise to topological inconsistencies or topological uncertainties.

In this paper we review the mathematical model of a map as a continuous surface with cell decomposition, and we examine a discrete-location/linear submodel which correctly models the computer's approximation to the continuous surface model. We describe the submodel, its relation to the larger model, special limitations of the submodel, and special useful properties of the submodel. In particular, we derive useful measures of stability of realizations of the submodel.

Because linear features in the submodel are straight line segments, the stability of the topology (or the robustness of the geometry) of a particular discrete-location/linear map realization turns out to be simply the upper bound distance that any line-segment endpoint may move in any direction and still not change the topological structure of the map. Looked at in the more general context, robustness is a geometric measure of the closeness of features on a map. Robustness is also a measure of the closeness of other maps having a one-to-one correspondence of point features, but having different topologies. This paper describes how to compute the geometric robustness of a particular geometric realization of a map and how to improve the topological stability. It also examines changes in stability that occur under various map update routines and transformation procedures. It proposes means of modifying or restricting those routines and procedures to preserve stability or to recover stability when it is diminished by those procedures.

INTRODUCTION

The Mathematical Model.

A standard or usual mathematical model for a map is the two-dimensional manifold or surface with a finite cellular decomposition. A two-dimensional manifold is a continuous, infinitely subdividable space such that every non-boundary point has a neighborhood that looks like a small disk in the plane, and every boundary point has a neighborhood that resembles a half disk. A cellular decomposition of a manifold is a partition of the space into mutually disjoint subsets, each of which is topologically equivalent to a point, an open interval, or an open disk (possibly with holes*). The partitioning subsets are called cells; and those cells which consist of a single point are called 0-cells; those which are topologically equivalent to an open interval are 1-cells; and the two-dimensional disks are called 2-cells. Partitioning means that every point in the map sheet belongs to exactly one of the cells in the finite collection. In other words, the union of the cells exhausts the space; and the cells themselves are pairwise disjoint. In order to guarantee that the cells do not overlap and that they fit together properly, the cells are defined in such a way that their boundaries do not belong to the cells themselves, but instead are made up of cells of lesser dimension. The cells must fit together with a plane-like smoothness and fill up the space. The rules for fitting together constitute the basis for the topological edits. Those rules are (1) combinatorial (i.e. describe finite relations among finite sets), and hence are machine-verifiable and (2) form a complete set of axioms for the theory of cellular structures on surfaces.

The Submodel.

A submodel of a mathematical model places additonal constraints on the model components, in our case the cells; and thereby, it reduces the number of legitimate instances of the model that must be considered.

The 1-cells in the usual topological manifold model are arcs or smooth curves. In practice, 1-cells are stored and displayed as polygonal lines, or polylines. The practice is based upon mechanical and mathematical constraints. Machines draw straight lines more easily than curved lines; and piecewise linear approximations are satisfactory approximations from a visual as well as a theoretical viewpoint. "Piecewise linear" is more suitable computationally for algorithm development; and piecewise linear can be as close as desired, certainly within machine precision constraints. For our submodel, we allow only polylines for our 1-cells.

** Some authors require that the 2-cells be simply-connected (no holes). This exposition does not. In fact, the structure of non-simply-connected surfaces with well-behaved singularities at the boundary is well known and is the basis for our understanding of our elementary 2-dimensional building blocks, the punctured 2-cells.*

The 0-cells in the usual topological model may take any real coordinate values on a surface that has infinite divisibility. In any implementation, however, machine precision will force the values into some finite grid. For our submodel, the 0-cells and the polyline interior vertices must have coordinates in some finite grid.

Our submodel puts considerable restrictions on the 0-cells and 1-cells; and one might ask if our submodel is as good as the general topological manifold for representing maps. In a very important sense, it is better for representing digital maps: Every computer implementation of a digital map is an instance or realization of our submodel; and many of the difficulties arising from machine precision constraints, such as topological uncertainty under transformation, can be better understood in the context of our finite-grid/ polyline (or discrete-location/linear) submodel.

While our 0-cells can come from only a finite set (in any particular instance, where the grid is given explicitly or implicitly), the points on our 1-cells are infinite in number. We keep track only of the vector ends of the segments making up the polylines; but our mathematical model requires that all of the points on a line segment be locatable, even though they cannot be explicitly stored.

Every instance of our submodel is also an instance of the more general topological manifold model; hence, we may use special properties of the submodel structure or use general properties of the larger model as needs arise. We examine topological stability in both contexts.

STABILITY

Continuous Deformation.

The mathematical notion of continuous deformation is just a formal representation of the intuitive concept. For a surface or manifold, S, in a space, K, a continuous deformation over time T is simply a continuous map:

$$\Phi: S\times[0,T] \dashrightarrow K$$

satisfying $\Phi(s,0) = s$, which says intuitively that, at time zero, every point is in its original position.

For each intermediate value of t in [0,T], we have the image of the ongoing deformation of S at time t given by:

$$\Phi(S\times\{t\}).$$

Continuous deformations need not preserve topological properties of S at each stage t. In other words, the intermediate image, $\Phi(S\times\{t\})$, may be topologically different from S. (It may even shrink to a single point if Φ is a contraction!) If S has a cell structure, then that cell structure may induce the same, a different, or no cell structure on the intermediate image, $\Phi(S\times\{t\})$.

We want to examine deformations that "almost always" preserve some cell structure on S (for all but finitely many values of t in [0,T]). Then we will be able to recognize when a deformation has changed the cell structure.

We also want to be able to distinguish small deformations from large deformations by looking at the distances through which the deformations move points. This is accomplished by limiting the maximum path length allowed in our deformations, where path length for each point s in S is the length of the arc:

$$\Phi(\{s\}\times[0,T]).$$

The class of all continuous deformations of our manifold is much too large to use for our study of stability. Moreover, this large class contains many exotic maps under which our cell structures become immediately unstable for all t > 0. In order to study stability, we examine families of deformations which do not move points too far and which move neighboring points in similar directions across similar distances. These deformations will be defined by their action on a finite set of points and extended in a piecewise linear manner to the whole space.

Our goal in this short paper is to study stability, not to develop a theory of interesting deformations. So without further elaborating on the theory behind the class of deformations described above, we simply point out that the deformations are defined for all instances of the larger continuous model and hence for all instances of the submodel. However, the intermediate image, $\Phi(S\times\{t\})$, of the deformation of an instance of the finite-grid/polyline submodel will not always be an instance of that submodel. Nevertheless, this intermediate image will always be an instance of the polyline submodel because of the piecewise-linear nature of the allowable deformations!

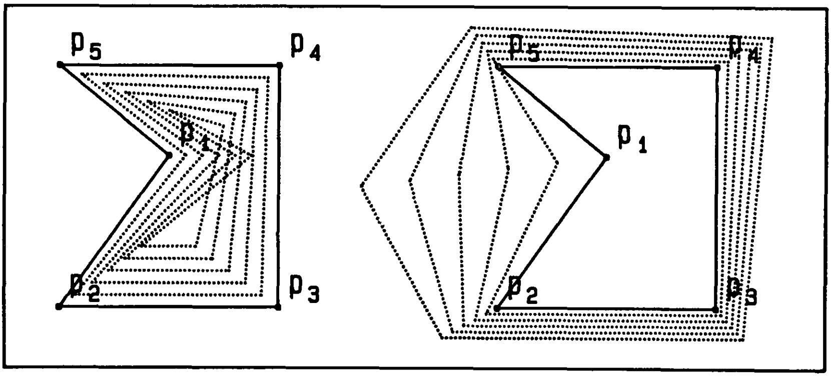


Figure 1. Illustrations of five intermediate deformations of polyline map portions

Notice in Figure 1 that the intermediate deformations on the left eventually change the cell structure when the lines double over on themselves. As the point p_1 moves to the right it gets closer to the linear feature p_3p_4, which itself is simultaneously moving to the left.

The initial polygon, however, becomes more stable if it is deformed as shown on the right. In the right-hand deformation, the features move toward an equilibrium position in which they are in some sense "as far from one another as possible." The "best" shape that they could attain in this simple example is a regular pentagon. The "good" deformation on the right is achieved by sending the vertices in just the opposite directions as in the "bad" deformation on the left.

The two deformations depicted in Figure 1 in some sense embody the basic ideas concerning stability:

(1) Stability is threatened when point features move toward nearby or nearest non-adjacent line segment features (and may then possibly cross over them!)

(2) Stability is improved when point features move away from nearby or nearest non-adjacent line segment features.

In the general situation of the 0-cells, 1-cells, and 2-cells of a map, however, the features are surrounded by other features; and movement is constrained in all directions:

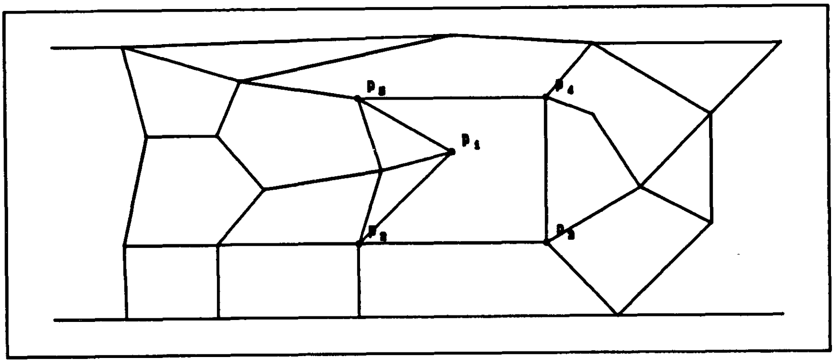


Figure 2. Cells in more complex map example.

In the example in Figure 2, the point p_1 is now further constrained by the additional features around it. That point is no longer free to move to the left to fill out the pentagon unless the points on the left of it move further to the left. There are two approaches that one may take with the general situation; and they correspond merely to assessing how good or bad the situation is, or to describing how to improve the situation. The easier first approach we will call "measuring robustness."

ROBUSTNESS

Robustness of a statistical estimator is a quality of permanence and reliability under varying conditions. We borrow the notion from the area of statistics, and we apply it to our geometric realizations of our map data. In statistics, an estimator is robust if it can withstand relatively large perturbations in the statistical data. For our application, a geometric realization of a specific cell configuration will be said to be a robust realization if it can tolerate considerable perturbation of feature positions without changing the cell structure.

If we ask how "bad" is the particular configuration, and where is it "worst," we may want to find one feature and the minimum distance we may perturb that feature (toward the nearest non-adjacent feature) to change cell structure. (Equivalently, we can ask for the least upper bound of distances that we can move all of the features simultaneously and still not change the cell structure.) If we ask how "good" can we make the map, we are asking the more difficult question of how to move all of the features simultaneously to a "best" or in some sense "most stable" position. That second problem appears to be much more formidable than the first, and rather like the classic unsolved n-body problem. We can, indeed, treat the problem as a force problem, and achieve interesting stability results. First, we will examine the easier problem of determining how unstable a feature configuration is and where the instability is worst by locating nearest non-adjacent feature pairs.

ROBUSTNESS AND INSTABILITY MEASURES

The following result regarding line segments is the key attribute that makes the finite-grid/linear submodel superior for studying instability.

(1) The minimum distance between two non-interesecting closed line segments is always attained by a pair of points, at least one of which is an end-point of one of the line segments.

This fact is easily seen and easily proved; however, the very important ramification of the fact is that computing distances between features in a polyline submodel boils down to computing point-to-line-segment distances, which are easy to compute.

If our topological data is stored in a TIGER-like file that "builds neighborhoods" in $O(N_f)$ time, where N_f is the number of cells in the neighborhood of a feature f, then the following algorithm will detect the nearest pair of features in $O(\Sigma(m^2))$ time where m is the polyline vertex count in each 2-cell, and the sum is over all 2-cells. The algorithm will also find the nearest segment to every point feature (0-cell or polyline vertex), and may be modified to yield nearest segment-to-segment distances using the fact (1) stated above.

Algorithm for computing robustness measures.

The minimum distance between a pair of features and the minimum distance from each vertex to a neighboring non-adjacent segment may be computed as follows:

INPUT: 0-cells, 1-cells, 2-cells, polyline vertex and polyline segment identifiers, and coordinates for 0-cells and polyline vertices.

OUTPUT: Closest-pair(a,b,c); where
a is a 0-cell or a polyline vertex identifier;
b is a polyline segment identifier; and
c is the distance between them.

Nearest-segment-to-x=(b,d); where
x assumes every 0-cell or a polyline vertex identifier;
b is the nearest polyline segment's identifier; and
d is the distance between them.

```
PROCEDURE NEAREST

Initialize Closest-pair(a,b,c) to any polyline vertex,
any polyline segment, and their distance.

FOR every 0-cell or 1-cell f DO

   Collect in a buffer all of the features that lie in
   the smallest closed neighborhood $N_f$ of f

   IF f is a 0-cell, THEN DO

      Initialize Nearest-segment-to-f to any
      non-adjacent segment and compute distance

      FOR each non-adjacent polyline segment in $N_f$ DO

         Compute distance to segment and update
         Closest-pair(a,b,c) and Nearest-segment-to-f,
         if necessary.

   ELSE DO

      FOR each interior polyline vertex v of f DO

        Initialize Nearest-segment-to-v to any
        non-adjacent segment and compute distance

        FOR each non-adjcnt. polyline segment in $N_f$ DO

           Compute distance to segment and update
           Closest-pair(a,b,c) and Nearest-segment-to-v,
           if necessary.

END PROCEDURE NEAREST
```

For most map inputs with polygons having relatively few components, the buffering step of collecting all features of N_f may be done in an array for faster processing.

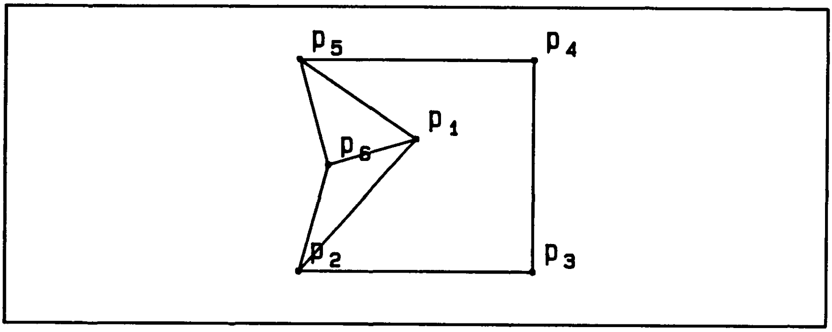


Figure 3. Smallest closed neighborhood of p_1.

RECOVERING STABILITY

Because the above algorithm examines every non-adjacent segment in the smallest closed neighborhood of every point feature (0-cell or interior polyline vertex), we may modify the algorithm to have it compute a net "force" of all of those non-adjacent segments on each point feature instead of having it merely locate the nearest segment, by making the following change:

Replace:

```
Compute distance to segment and update
Closest-pair(a,b,c) and Nearest-segment-to-f (or v),
if necessary.
```

By:

```
Compute the force on point feature due to segment
and add to net-force-on-f  (or v).
```

Since the non-adjacent segments of the smallest closed neighborhood surround the point feature and in some sense isolate the point from effects of other segments, it makes sense to use this force model. As with the n-body problem, we can compute a force on each of our vertices in our initial configuration. We may model the forces on a vertex to be inversely proportional to the distance of points on neighboring non-adjacent segments.

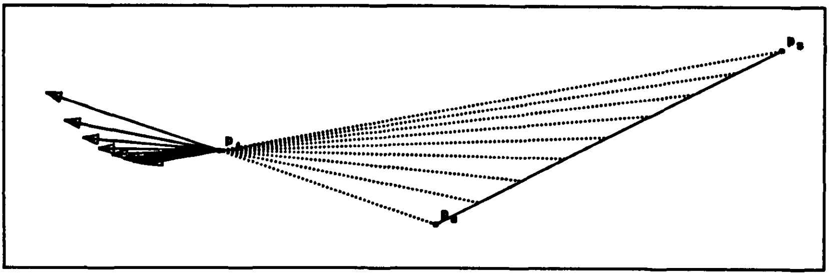


Figure 4. Forces exerted on a point by a line segment.

We may sum the forces by a straightforward vector integration. The result will be a vector whose magnitude and direction provide a best initial direction and speed to move our vertex in order to improve stability. As with the n-body problem, computing initial forces is not difficult. The hard part is determining the movement of the system, and, in our case, finding the eventual equilibrium position. We may simulate the movement by iterative linear approximations; and, perhaps surprisingly, that approach looks promising.

FUTURE DIRECTIONS

The usual drawbacks to iterative methods are cost and convergence. Some experimentation is required to learn more about convergence, but our finite grid submodel promises to be extremely useful in establishing a bound for tolerances to replace "exact or total stability."

Cost also remains managable. Because the force computation is local, depending only on the smallest closed neighborhood, N_f, we can achieve, for all size maps having approximately the same local neighborhood configurations, a linear (in the number of point features) force computation algorithm. This possibility makes an iterative approach to stability improvement seem reasonable for large maps as well as for small maps. We plan to do more experimenting with iterative approaches to stability improvement.

REFERENCES

Aho, A. V., J. Hopcroft, and J. Ullman, 1983, Data Structures and Algorithms, Addison-Wesley, Reading, MA

Corbett, James, 1979, Topological Principles in Cartography, Bureau of the Census Technical Paper 48.

Dugundji, James, 1966, Topology, Allyn and Bacon, Inc., Boston.

Guibas, Leonidas and Jorge Stolfi, 1985, "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," ACM Transactions on Graphics, Vol. 4, No. 1 (April), pp 74-123.

Pavlidis, Theo, 1982, Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, MD.

White, Marvin, and Patricia Griffin, 1979, "Coordinate Free Cartography," AutoCarto IV, Vol II, Proceedings of the Fourth International Symposium on Cartography and Computing, pp. 236-245.

White, Marvin, 1984, "Technical Requirements and Standards for a Multipurpose Geographic Data System, The American Cartographer, Vol. 11, No. 1, pp. 15-26.

HIPPARCHUS GEOPOSITIONING MODEL: AN OVERVIEW

Hrvoje Lukatela
33 Chancellor Way, Calgary, AB T2K 1Y3 Canada

ABSTRACT

This paper introduces a novel digital geopositioning model. It is based on computational geodesy in which direction cosines are used instead of the conventional -angular- ellipsoid normal representation, and eccentricity term expansions are replaced by iterative algorithms. The surface is partitioned by a spheroid equivalent of the Voronoi polygon network. The model affords seamless global coverage, easily attainable millimetric resolution, and data-density sensitive location indexing. Numerical representation of 0, 1 and 2 dimensional objects is in complete accordance with the unbounded, spheroidal nature of the data domain, free from any size, shape or location restriction. Efficient union and intersection evaluations extend the utility of the relational technique into the realm of geometronical systems with non-trivial spatial precision requirements. Digital modelling of orbital dynamics follows closely the numerical methodology used by terrestrial geometry. The HIPPARCHUS software package includes the transformations and utility functions required for efficient generation of transient graphics, and for the communication with systems based on conventional cartographic projections.

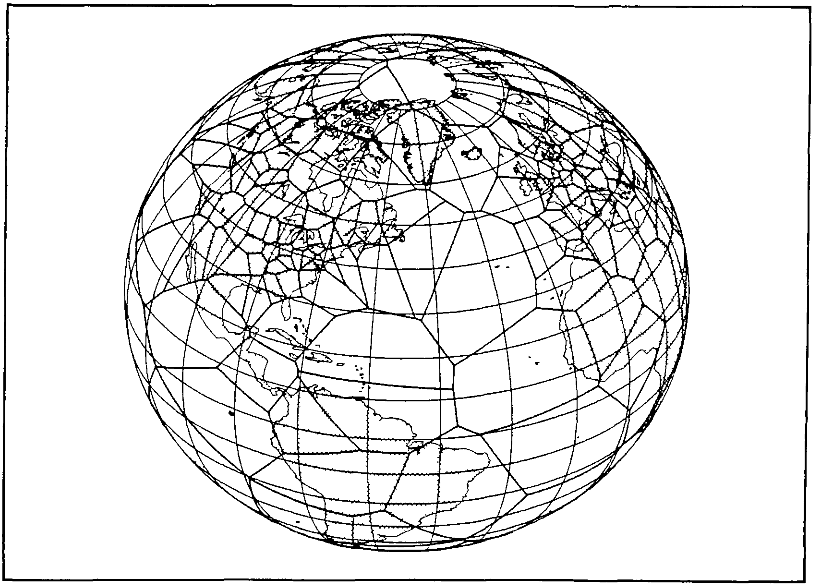


Fig. 1: HIPPARCHUS ellipsoid surface partitioning scheme

INTRODUCTION

A numerical geopositioning model is an essential element of any system wherein a dimension of space enters into the semantics of the application - and therefore into the software technique repertoire - in a fundamental way. It consists of location attribute data definitions and computational algorithms, which allow position sensitive storage and retrieval of data, and provide a basis for evaluation of spatial relationships. (The term "spatial relationship" is used in this paper to describe the formal statement of any practical spatial problem which deals with positions of real or abstract objects on - or close to - the Earth surface. Their nature can vary; examples might include geodetic position computations, course optimization for navigation in ice-infested waters or determination of the most probable location of objects remotely sensed from a platform in the near Space.)

If the location attributes of data elements in a computer system are used exclusively for the generation of a small-scale analog map document, the demands made of a geopositioning model are few and simple. When the area of coverage is limited, and projection geometry, spatial resolution and partitioning of the data can be made directly compatible with same characteristics of all future required products, a single plane coordinate system is often employed. Such a system is usually based on one of the large-area conformal projections (e.g. Lambert, Gauss-Krueger, etc.), and provides adequate means to identify positions, partition the data, and construct a location index. The model may even allow limited spatial analysis.

However, with the increase of the area of coverage and the functional power of information systems, the nature of the problem changes considerably.

Precision requirements usually exceed the level of difference between planar coordinate relationships and the actual object-space geometry. In most cases, the generation of an analog map is reduced to a secondary objective. Location attributes are primarily used to support the evaluation of spatial relationships required by the application. Indeed, as the volatility and volume of data grows, it becomes increasingly common that a location-specific item enters a system, contributes to the evaluation of a large number of spatial relationships, and is ultimately discarded, without ever being presented in the graphical form.

Even in systems used primarily to automate the production of analog documents, there is often a need to accommodate many different projection, resolution and data partitioning schemes on a continental or even global scale.

A point is thus quickly reached where geopositioning model must satisfy very demanding functional requirements, yet any restriction on the data domain becomes unacceptable. From the application point of view, the mapping from an atomic surface fraction into a distinct internal numerical location descriptor must be global, continuous and conjugate.

Faced with these requirements, manual spatial data processing resorts to a combination of two techniques. A set of multiple planar projection systems (e.g. UTM "zones") is used to achieve - seldom successfully - the global coverage. Initially simple calculations are cluttered with various "correction" terms in order to deal with differences between planar coordinates and true object geometry.

A failure to understand the precise nature of spatial data (especially, by ignoring the profound conceptual difference between an analog map and the true data domain) often leads to a blind transplant of conventional cartographic techniques into a computerized system.

This seldom results in a satisfactory geopositioning model: cartographic projections are notorious for their computational inefficiency; global coverage usually requires the use of location-specific transformations. Programming becomes progressively more complex as the precision requirements increase. Boundary problems are difficult to solve; this imposes discontinuities or size restrictions for the models of spatial data objects. Finally, classical cartography offers little or no help in modelling of the near-space geometry. The same system can therefore be forced to employ two disparate numerical methodologies: one for the positions on the Earth surface and quite another for orbital data. This presents an increasingly serious problem in many emerging high data-volume applications.

Design (or selection) criteria for a generalized location referencing numerical model and software will change from one computerized information system to another, but will be based - usually - on the size of the area of interest, spatial resolution, anticipated data volume, optimal computational efficiency, logical and geometrical complexity of objects modelled, and on the level of precision with which all these elements can be defined before the system is built. Nevertheless, it is possible to list important functional requirements that will pertain to a majority of extended coverage geographic information systems:

- Unrestricted numerical representation of arbitrarily-sized and -shaped objects with 0, 1 and 2 dimensions (i.e. points, lines, regions) relative to the surface of the Earth, and efficient evaluation their unions and intersections.

- Global coverage, without any regions of numerical instability or deterioration; ability to precisely model spatial relationships resulting from the unbounded, spheroidal nature of the data domain.

- Variable (application controlled!) levels of positional resolution and computational geometry precision; up to sub-millimeter level for location framework or field-measurement related data.

- High utilization level of the coordinate data-storage space.

- Construction of data density and system activity level sensitive surface partitioning and indexing scheme; capability of dynamic re-partitioning in order to respond to a change in density or activity pattern of an operational system.

- Ability to effectively model the time/space relationships of surface, aeronautical and orbital movements.

The quality of a generalized geopositioning model will obviously depend not only on the extent to which the above criteria have been satisfied, but on its software engineering potential as well. The model must be capable of being implemented in program code which is efficient, reliable, portable, and easily interfaceble to a large number of different types of data-access services (i.e. file and indexing schemes, database software packages e.t.c.) and application problem-solving programs.

The geopositioning model presented here consists of three key components: a) spheroidal cell structure analogous to planar Voronoi polygons; b) computational geodesy based on closed iterative algorithms, and c) an unlabored representation of global ellipsoid coordinates in terms of a cell identifier and description of location within the cell. Since the computational bridge between the global position and the location within the cell consists of a pseudo-stereographic ellipsoid-plane transformation, HIPPARCHUS has been chosen as the name for the model. (Hipparchus, (180-125 B.C.) - inventor of stereographic projection: the first truly practical geopositioning model.)

The HIPPARCHUS model provides a unique spatial framework, and includes the algorithms necessary to encode data and evaluate spatial relationships. In doing so, it attempts to satisfy - to the highest extent possible - all the requirements mentioned above. The nature of the framework and principles of its data manipulation techniques will be examined next in some detail.

GLOBAL ELLIPSOID COORDINATES

A plane or sphere can be used to represent the surface of the Earth only for limited-area, low-precision computations. A general purpose geopositioning model will, however, require a better fitting surface. Typically, a quadric, biaxial (rotational) ellipsoid is employed. (Triaxial ellipsoid and various sets of polynomial correction terms to a biaxial ellipsoid have both been employed in geodetic calculations and proposed for general cartographic use. The discussion of potential merits of those surfaces, and the ability of the proposed model to accommodate them numerically, are beyond the scope of this text.) The parameters of size and eccentricity of the reference ellipsoid can be determined by a combination of theoretical investigation into the equilibrium shape of a rotating near-liquid body and terrestrial geometry and satellite orbit observations. This is an open-ended process, resulting in occasional corrections of ever-decreasing magnitude.

The position on the surface of the ellipsoid can be represented numerically in many different ways. Conceptual clarity of the model, as well as practical software engineering considerations, demand that one such representation be used as a canonical form of global location descriptor throughout the model. The selection of this numerical form is one of the most critical decisions in the design of a geopositioning model.

The traditional angular measurements of latitude and longitude are extremely unsuitable for automated computations. Few, if any, spatial problems can avoid multiple evaluations of trigonometric functions. Moreover, convoluted programming techniques are often necessary to detect areas of numerical instability and adjust an algorithm accordingly. It would be simple to use Cartesian point coordinates instead, but the domain would no longer be restricted to the ellipsoid surface. An additional condition would have to be incorporated into the statement of most surface-related geometry problems.

The geometrical entity described by latitude and longitude is a vector normal to the surface of ellipsoid in the location thus defined. This vector can be expressed by its direction cosines, and a normalized triplet can be used as coordinates of a surface point. This appears to be an ideal canonical location descriptor: the domain is restricted to the surface; numerical manipulations based on vector algebra productions are easy to program and simple to test, and a common 64-bit floating point numbers will yield sub-millimeter resolution even at radial distances that are an order of magnitude above the surface of the Earth.

Conventional formulae for the solution of ellipsoid geometry problems were typically obtained by expansion in terms of an ascending power series of eccentricity. While this was unavoidable for problems lacking a closed solution, it was also often used in order to reduce the number of digits which had to be carried in a numerical treatment of a geodetic problem with a limited spatial extent. As long as the eccentricity of the reference surface was constant, any a priori precision criterion could be satisfied by either finding the maximum value of the remainder dropped, or - more commonly - by deciding on the threshold exponent beyond which terms could be ignored for a whole class of practical problems.

Formulae thus obtained are useful for manual calculations but do not provide a sound base for the construction of efficient and data-independent computer algorithms.

The insight required to decide whether or not a particular set of formulae can or can not be used to solve a given problem is difficult to replicate in a program. Expansions must be checked and programmed with extreme care, since the influence of errors in higher terms can be easily mistaken for unavoidable numerical noise in the system. While the assumption of moderate and constant elliptical eccentricity might be valid for terrestrial problems, it represents an undue limitation in systems incorporating orbital geometry. Finally, in most computer hardware environments the full number of significant digits required to achieve sub-millimeter resolution can be used without any penalty in the execution time.

With the appropriate statement of conditions, all ellipsoid geometry problems of single periodic nature (i.e. those whose differential geometry statement does not lead to elliptic integrals) can be solved very efficiently, to any desired level of precision, using an iteration technique based on the alternate evaluation of conditions near the surface and at the point where the normal is closest to the coordinate origin. Ellipsoid coordinates consisting of three direction cosines offer significant advantages in all numerical algorithms required to carry out this iteration. The distinct advantage of this method (compared to a program based on expansion formulae) lies in its automatic self-adjustment to the computational load. The number of iterations will depend on the precision criterion, physical size of the problem and the measure of ellipsoid (or ellipse!) eccentricity. The same program can therefore be used for all global geometry problems of a given type, with full confidence that the desired precision has been achieved - in each individual invocation - through a minimum number of arithmetical operations necessary.

This approach can be applied not only to conventional geodetic problems but also to solve problems dealing with both surface and spatial entities. In particular, it will be effective solving the problems which deal simultaneously with the ellipsoid surface and with orbital parameters which are themselves of quadric nature.

It should be noted that only the framework data must be permanently retained in global ellipsoid coordinate values. As explained below, volume data coordinates can be stored in a much more efficient format, and transformed into ellipsoid coordinates in transient mode, whenever these are required.

SURFACE PARTITIONING AND LOCATION INDEXING

One of the essential facilities required for the design and construction of a geographical database is a surface partitioning scheme. On the simplest level, this provides a basis for indexing and retrieval of location-specific data. Even more important will be its use for efficient run-time evaluation of spatial unions and intersections, probably the most critical facility in construction of a fully relational spatial database system.

Where the potential for extended coverage is required, the partitioning scheme must be capable of dealing with the complete ellipsoidal surface. This can not be achieved using any of the regular tessellations which have been proposed as the base for hierarchical data-cells: beyond the equivalent of the five Platonic solids, the sphere can not be divided into a finite number of equal, regular surface elements.

Various schemes based on latitude/longitude "rectangles" are often used for large coverage or global databases. However, resulting cell network is hard to modify in size and density, high-latitude coverage can be restricted or inefficient, and in most cases the approach forces the use of unwieldy angular coordinates.

By contrast, the partitioning scheme used in the HIPPARCHUS model is based on spheroidal cells analogous to planar Voronoi polygons. The definition of the structure is simple. Given a set of distinct (center)points, a spheroidal polygon-cell corresponding to one of them is defined as a set of all surface points "closer" to it than to any other member of the centerpoint set. For each surface point, the minimum "distance" to any point in the set of centers can be determined: if there is only one centerpoint at such a distance, the point is within a cell. If there are two, it belongs to an edge. If there are three, the point is a vertex. A dual of the set of polygons is obtained by connecting the centerpoints which share an edge.

The application can define a pattern of cells by any purposefully distributed set of centerpoints. Since these are defined by their normals, the partitioning scheme is completely free from condescending to any numerically singular surface point. The distribution of centerpoints can be based on any combination of criteria selected by the application: data volume distribution, system activity patterns, maximum or minimum cell size limits. It can even represent an existing set of spatial framework items, e.g. geodetic control stations.

A sort-like algorithm produces the digital model of the dual. The cell frame structure is thus reduced to a list of global, ellipsoid coordinates of centerpoints and a circular list of neighbor identifiers for each cell. If the application requires that a limit be placed on the maximum "distance" between neighboring centerpoints, the algorithm must be capable of bridging the "voids", and null items must be recognizable in the circular list. This data structure is used extensively by all spatial algorithms. Unlike systems in which location of the cell is implied in its identifier, the HIPPARCHUS model requires explicit recording of the global coordinates of cell centers. Method of storage and access to this data can therefore have considerable influence on the efficiency of spatial processing.

A cell is assigned an internal coordinate system with the origin at its centerpoint. As mentioned before, the mapping function between global and cell systems is an ellipsoid-modified stereographic projection. The "transformation algorithms" (in both directions) consist therefore of nothing but a few floating-point multiplications.

"Finite Element Cartography". If a large volume of data has to be transformed into output device coordinates based on a specific conventional cartographic projection, only a few points on the cell (or the display surface) frame will have to be transformed using a rigorous cartographic projection calculation. Based on the frame data/display correspondence, parameters of a simple polynomial transformation are easily calculated. Volume transformations will again require only a few multiplications, and can be set to produce the result directly in hardware coordinates of an output device. This type of manipulation can be of particular value if a complex geometronical function has to be applied over the complete surface of a dense data set, for instance in transient cartographic restitution of digital remote sensor image material.

One of the most often executed algorithms in the model will probably be the search for the "home cell" of an arbitrary global location. Selection of the first candidate cell is left to the application, in order to exploit any systematic bias in either transient or permanent location reference distribution. A list of all neighbors is traversed, and distances from the given location to the neighbor centerpoints are determined. If all these distances are greater than the distance from the current candidate centerpoint, the problem is solved. Otherwise, the minimum value indicates a better candidate. While the algorithm is very straightforward, its efficiency will be extremely sensitive to the selection of the spheroid "distance" definition and numerical characteristics of global coordinates. The same will apply to most combined list-processing and numerical algorithms employed by the model.

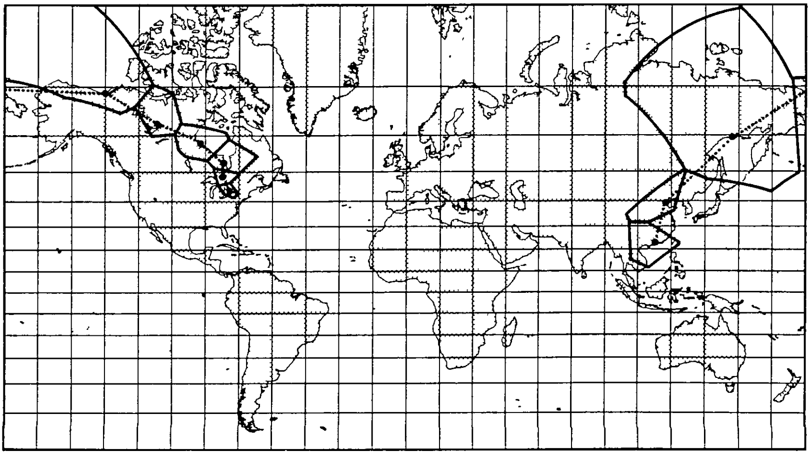


Fig. 2: Trace of home cell search algorithm

While Voronoi polygons have often been used in computer algorithms solving various classes of planar navigation problems, at the time of this writing no record was found of the use of an equivalent global, spheroidal structure as a partitioning scheme in a geometronical computer system.

MODELLING OF SPATIAL DATA OBJECTS

Points: Digital representation of a point data element is simple: it consists of a cell identifier and local (cell) coordinates. Even with fairly large cells, the global-to-local scaling will ensure equivalent spatial resolution in case where local coordinate values have only one-half of the significant digits used for global coordinate values. Since the efficiency of external storage use and the associated speed of I/O transfer can be of extreme importance in a large database, the following numerical data are of interest:

If a 64-bit global, a 32-bit local coordinate values and 16-bit cell identifier are used, the volume data point representation will require only 80 bits, and will still yield sub-millimeter resolution. 80 binary digits are capable of storing 2**80 (approximately 1.2E24) distinct values; the surface of the Earth is approximately 5.1E20 square millimeters. The ratio of these two numbers (approximately 69 out of 80) represents the theoretical memory utilization factor; practically, the margin allows significant variation in cell size and use of various computational conveniences (floating point notation, cell range encoding, e.t.c.). This utilization factor compares to 69 bits out of 128 if the point is represented by latitude/longitude in radian measure, and 69 out of 144 bits (typically) if a conventional, wide-coverage cartographic projection system plane identifier and coordinates are used. Furthermore, various external storage compression schemes that take advantage of the re-occurring cell identifier are likely to be significantly simpler and more effective than any compression scheme of a pure numerical coordinate value.

It is important to note that in HIPPARCHUS model cell coordinates of a point are not used for a numerical solution of metric problems; their purpose is to provide a compressed coordinate storage format for high-volume data, and to facilitate generation of the transient, analog view of the data.

Lines: One-dimensional objects are represented by an ordered list of cells traversed by the line, and - within each cell - a list, (possibly null) of vertices in the point format described above. If the application requires frequent evaluations of spatial unions and intersections, it might be efficient to find and store permanently all points where lines cross cell boundaries. Their internal representation (permanent or transient) is somewhat modified in order to restrict their domain to the one-dimensional edge, but their resolution and storage requirements will be comparable to the general point format used by the model.

Regions: Two-dimensional objects are represented by a directed circular boundary line and an encoded aggregate list of cells that are completely within the region. When compared to simple boundary line circular vertex list, this structure makes the evaluation of spatial relationships significantly more efficient. The solution will often be reached by simple manipulation of cell identifier lists, instead of the evaluation of boundary geometry. The number of cases where, ceteris paribus, this will be possible, will be inversely proportional to the average cell size. (In example in Fig. 3, boundary geometry examination will be confined to three cells.) This representation of a two-dimensional object is a combination of the traditional boundary representation and schemes based on regular planar tessellations. It offers the high resolution and precision usually associated with the former, while approaching the efficiency of relational evaluations of the latter. In addition, it does not violate the true spherical nature of the data domain. For instance, if [A] is a region, then NOT [A] is an infinite, numerically ill-defined region in a plane. By contrast, on any spheroidal surface NOT [A] is the simple finite complement.

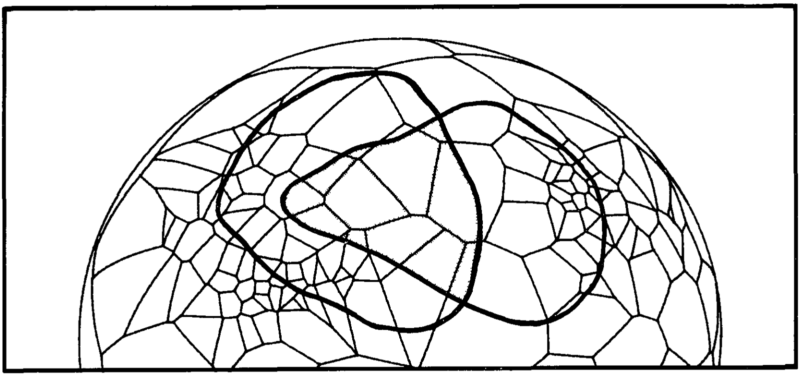


Fig. 3: Intersection of two-dimensional objects

Orbit Dynamics: Practice abounds with examples of problems encountered in attempts to integrate remote sensing and existing terrestrial data. Even in instances where the spatial geometry can be defined with sufficient precision, it is common to cast (by "pre-processing") the digital image produced by a satellite sensor into a specific plane projection system and pixel aspect ratio and orientation. This unnecessarily increases the entropy of remotely sensed data available to applications requiring different or no planar castings. In many instances, problems will disappear if the application is given the ability to manipulate the original, undistorted, observation geometry.

A general-purpose geopositioning software tool must therefore provide efficient evaluation of basic time/geometry relationships within the orbital plane, and the ability to transfer the locations from an instantaneous orbit plane to its primary frame of spatial reference. (More complex calculations are probably application-specific and are restricted to infrequent adjustments of orbit parameters.)

The geometry functions described already suffice to define any orbit at the convenient epoch - e.g. the time of the last parameter adjustment. To find a position (in the orbital plane) of a platform at a given time, a direct solution of the problem postulated by Kepler's second law is required. (Same as in geodetic problems mentioned previously, this "direct" problem requires an iteration, while the "inverse" yields a closed solution.) Any increase of orbit eccentricity will affect the number of iterations, but the same software component can be used to solve both near-circular and steep orbits. Common 64-bit floating point representation will preserve millimetric resolution even for geosynchronous orbits. Rigorous modelling of general precession can be achieved simply by an additional vector rotation about the polar axis. This is combined easily with sidereal rotation, required in any case for transfer of position between the inertial and terrestrial frames of reference.

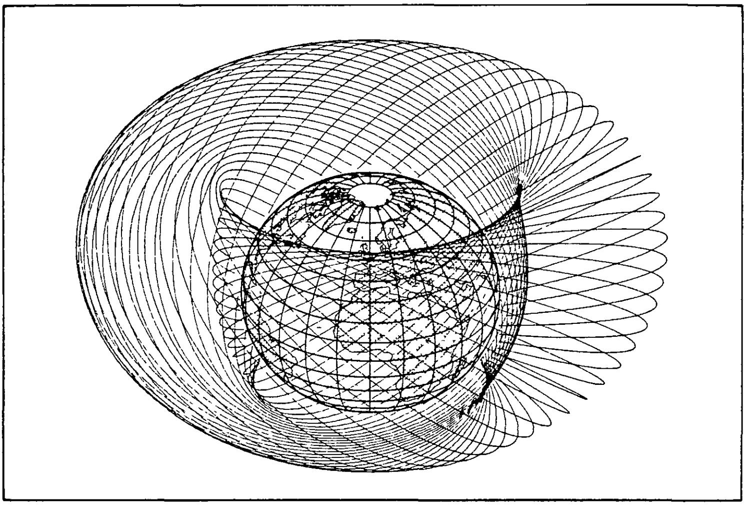


Fig. 4: Orthographic view of a precessing orbit

CONCLUSION

Use of computers in mapping is as old as the computer itself: the first commercially marketed computer, UNIVAC 1, was used in 1952 to calculate Gauss-Krueger projection tables. With the development of computer graphics, it quickly became common to store and update a graphical scheme representing a map. Until very recently, the main object of this process remained the production of graphical output that was not substantially different from a conventional analog map. While the production of the map was thus computerized, the ability of an "end-use" quantitative discipline to employ a computer to solve complex spatial problems was not addressed. The use of a "computer map" was precisely the same as that of a traditional, manually produced document.

All quantitative disciplines are facing the same demands as cartography to increase precision, volume and complexity of data which can be efficiently processed. Hence, computer applications in those disciplines require "maps" from which spatial inferences can be derived not only by the traditional map user, but also by a set of computer application programs. To a limited extent only, this has been achieved in applications which could tolerate severe limitations on area of coverage, data volumes, or spatial resolution requirements. Location attributes in these computer systems are usually based on an extended coverage ellipsoid-to-plane conformal projection: a numerical model developed for a completely different purpose.

Computer systems requiring extensive spatial modelling combined with high resolution and global coverage need powerful yet efficient numerical georeferencing models. It is unlikely that these can be based on conventional cartographic techniques. Numerical methodologies designed specifically for the computerized handling of spatial data have the best potential for providing generalized solutions.

A POLYGON OVERLAY SYSTEM IN PROLOG

Wm. Randolph Franklin
Peter Y.F. Wu
Electrical, Computer, and Systems Engineering Dept.
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
U.S.A.
(email: franklin@csv.rpi.edu,wup@csv.rpi.edu)

ABSTRACT

A system for polygon overlay is developed using Prolog. The algorithm expands the concept of local processing and decomposes the process of polygon overlay into a number of stages, resulting in much simplified data structures. The system in Prolog adopts a relational approach to data structuring. Geometric entities are defined as Prolog facts, and the Prolog rules encoding geometry algorithms perform data processing. Processing follows a paradigm of set-based operations using iterative search and backtracking. Calculation of geometric intersection is done using rational arithmetic. Numerical accuracy is therefore preserved, and hence the topological consistency is guaranteed. Special cases of chain intersection, that is, touching and partially overlapping chains, are handled properly. A strategy to remove sliver polygons due to coincidental input is outlined.

INTRODUCTION

This paper presents a polygon overlay system developed in Prolog. We decompose the process of polygon overlay into a number of stages, each of which performs certain local operations. Our strategy simplifies data structuring. Furthermore, we achieve stability using rational arithmetic to compute geometric intersections.

Prolog offers several advantages as a programming tool for geometry algorithms. A fundamental problem in dealing with geometry on the computer is the primitive nature of conventional programming languages. Conceptually simple ideas often are unexpectedly difficult to implement. The descriptive nature of Prolog provides a much more intuitive programming environment, and hence fosters more readable programs. More specifically, Prolog rules are well suited to coding geometry algorithms for set-based operations (Swinson 82,83; Gonzalez 84; Franklin 86). Inherent to the problem of implementation is the design of data structures. Decomposing the complicated process of polygon overlay, we have simplified the data structures. Furthermore, Prolog provides a built-in relational database which makes data structuring even easier. Another problem in the implementation of geometry algorithms is numerical inaccuracy which results in topological inconsistency. The problem stems from discretization errors in finite precision computer arithmetic (Franklin 84). Prolog has the flexibility of operator overloading to allow modular installation of other arithmetic domains, such as rational numbers, in existing programs. Our system for polygon overlay makes use of this feature.

The purpose of this paper is therefore two fold: to demonstrate that Prolog is a viable programming tool for geometric and geographic data processing, and to present rational arithmetic as a practicable solution to the problems stemming from discretization errors, in polygon overlay. In this paper, we will first briefly survey the previous works on the polygon overlay problem. Then, we will describe our polygon overlay system and the design of our data structures. We will discuss further on the issue of using rational arithmetic to calculate geometric intersections, and will outline an approach to removing slivers due to coincidental input data. We have implemented our system on a SUN 2/170 machine running C-Prolog version 1.5 (Pereira 86) and we are gathering more results at the time of this writing.

A BRIEF SURVEY

Polygon overlay encompasses a number of geometric and topological computation problems. During the 70's when geographic information systems were first developed, geographers thought of polygon overlay as "the most complex problem of geographic data structuring..." (Chrisman 76). Reports studying the problem at further length also had similar remarks (Goodchild 78; White 78). Guevara presented formal treatment and an analysis of several solutions in his thesis (Guevara 83).

Fundamental to the development of a solution during the 70's was the research effort in computational geometry. Solutions to basic problems of polygon intersection, and point-in-polygon inclusion were reported in (Eastman 72; Franklin 72) and (Ferguson 73), respectively. Shamos and Bentley in 1976 developed a number of algorithms to efficiently solve many geometry problems (Shamos 76). Preparata and Shamos organized most of the work in the design and analysis of geometry algorithms in their book (Preparata 85). Algorithms in determining polyline intersections were particularly important to polygon overlay. Burton designed a data structure, called Binary Search Polygon Representation, for efficient processing of polygons and polylines (Burton 77). Study on algorithms to determine polyline intersections were reported in (Freeman 75) and (Little 79).

Systems with polygon overlay implemented were available in the late 70's. The well-known CGIS - Canadian Geographic Information System combined grid/raster based approach with vector based approach to perform map overlay (Tomlinson 76). Two systems, PIOS - Polygon Information Overlay System (DeBerry 79) and MOSS - Map Overlay and Statistical System (Reed

82), both operate on two polygons at a time in pairwise comparison. By far a much more advanced algorithm due to White introduced the concept of local processing in WHIRLPOOL - a program in the system ODYSSEY (White 78). Franklin described an adaptive grid for efficient determination of intersecting objects (Franklin 80,83a). We implemented this idea in Prolog as presented in this paper. Teng reported a system taking a topological approach which quite likely is based on the concept of local processing (Teng 86).

DESCRIPTION OF THE SYSTEM

Polygon overlay is the process of superimposing two maps: Given two maps A and B, the polygon overlay process produces an output map C which comprises all the information of each input map, as well as the spatial correlation information. Map C contains all the chains of A and B; intersecting chains are split at the intersection points. Thus C contains all the nodes of A and B, and the new nodes generated at the intersection points. Each polygon in C is the intersection of two polygons, one in each of A and B.

Input/Output File Structures

We assume data consistency in our input maps. An input map is a file of variable length records. Each record is a Prolog "fact" in the following format:

chain(C,N1,N2,[[X,Y],..],P1,P2)

Each record is uniquely identified by name *C*; *N1*,*N2* are the names of the beginning and ending nodes; [[*X*,*Y*],...] is the list of (x,y) coordinates for the vertices along the polyline structure from *N1* to *N2*; and *P1*,*P2* are the polygons to the left and right of chain in the direction from *N1* to *N2*. The output map is a set of chain records in the same format, with each polygon identified as the intersection of two input polygons.

An Overview of The System

Our system divides the polygon overlay process into three major stages. Each is further subdivided into a number of steps. The following presents a brief description of the three stages. We will then discuss each stage in further detail.

1. *Chain Intersection.*
   Determine intersecting chains and split them at the intersection points.

2. *Polygon Formation.*
   Link the chains to form the output polygons.

3. *Overlay Identification.*
   Identify each output polygon as the intersection of two input polygons.

Chain Intersection Determining chain intersections is inevitably the performance bottle-neck in the polygon overlay process. To speed it up, we cast an adaptive grid over the edge segments (Franklin 83b). The intention is to isolate cases of intersection to within those elements that occupy the same grid cell. Implemented in our polygon overlay system, it involves the following steps:

1.1 Compute an appropriate grid size to form the grid.

1.2 Cast each edge element into each of the grid cells occupies. For each edge segment *E*, enter a fact *edge_in_grid(E, G)* for each grid cell *G* occupied by *E*.

1.3 Collect the edges in each grid cell for pairwise comparison. For each grid cell with potentially intersecting edges *E1,E2,...*, enter a fact *edges_in_same_grid(G,[E1,E2,...])*.

1.4 For each grid cell, test all pairs of edges in it to determine intersecting pairs. Split the edges and form a new node at the intersection point.

Figure 1 depicts the casting of a grid to isolate the intersection cases. The process splits the intersecting chains at the point of intersection. Hence, we have all the chains of the output map: the chains do not intersect each other except at the nodes.

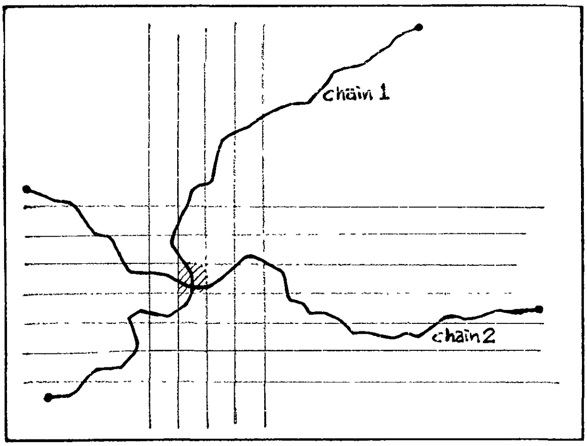


Figure 1. The grid isolates potentially intersecting edge segments.

Polygon Formation. Here we connect the chains to form polygons of the output map. The steps involved are the following:

2.1 For each chain, identify the incident nodes at both head and tail, and calculate the incident angles. Enter the facts as *incidence(node,chain_incidence,angle)*.

2.2 At each node, sort the incident chains into proper cyclic order. For each node *N*, enter the facts as *node(N,[C1,C2,...])* for chain incidences *C1,C2,...* sorted in order.

2.3 Each consecutive pair of incident chains identifies a corner of an output polygon; enter the facts as *linkage([C1,C2])*, *linkage([C2,C3])*, . ., *linkage([Cn,C1])*, one for each pair of adjacent chains.

2.4 Link up the linkage facts in proper cyclic order. For example, connect *linkage([C1,C2])* and *linkage([C2,C3,C4])* to form *linkage([C1,C2,C3,C4])*. Each complete cycle, such as *linkage([C1,C2, . ,C1])* identifies a polygon.

Figure 2 illustrates these steps. We form all the polygons with only local operations.

Overlay Identification. For each output polygon, we need to determine the two polygons, one from each input map, which intersect to form it. We observe that there are two kinds of output polygons: If the boundary chains of output polygon *C* involve the chains of two different polygons *A* and *B*, one from each map, *C* is $A \cap B$. Otherwise, all the boundary chains around *C* must come from the same polygon, *A* which is completely contained in a polygon *B* in the other input map. Then *C* is $A \cap B$. To determine *B*, we can search the neighbors of *C* and their neighbors, and so on. The search fails only when the two input maps are not involved in any chain intersection.

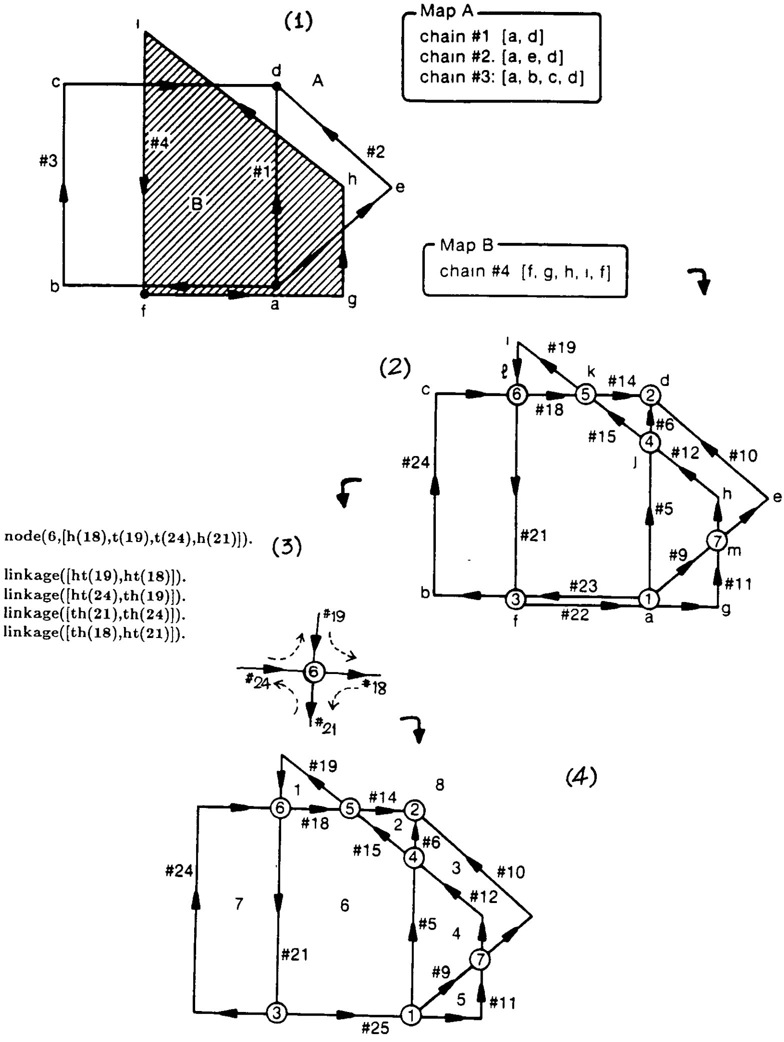
EXACT RATIONAL ARITHMETIC

Rational arithmetic has been in use in many symbolic mathematics computation systems, most notably MACSYMA (Macsyma 83), which is in Lisp. The Unix system also provides a library of multiple precision integer arithmetic in C (Sun 86a), and serveral tools are available for calculations using rational arithmetic (Sun 86b). We developed a package for exact rational arithmetic in Prolog (Wu 86). In exact rational arithmetic, we evaluate an expression to a fraction, of both denominator and numerator as integers with virtually no overflow limit. Since Prolog allows operator overloading, the syntax for arithmetic expression remains unchanged. Installation of the package in an existing program is relatively simple.

Stability and Special Cases

Numerical inaccuracy has long been a problem with geometric computation, since it leads to topological inconsistency. With rational arithmetic we are able to circumvent problems of arithmetic inaccuracy arising from

(1) polygon overlay example
(2) nodes and chains after splitting intersecting chains
(3) form linkage record for each polygon corner at every node
(4) connect linkages to form polygons



Map A
chain #1 [a, d]
chain #2. [a, e, d]
chain #3: [a, b, c, d]

Map B
chain #4 [f, g, h, i, f]

```
node(6,[h(18),t(19),t(24),h(21)]).

linkage([ht(19),ht(18)]).
linkage([ht(24),th(19)]).
linkage([th(21),th(24)]).
linkage([th(18),ht(21)]).
```

```
polygon(1, [ht(18), ht(19)])
polygon(2, [ht(6), th(14), th(15)])
polygon(3, [th(6), th(12), ht(10)])
polygon(4, [th(5), ht(9), ht(12)])
polygon(5, [ht(11), th(9)])
polygon(6, [ht(5), ht(15), th(18), ht(21), ht(25)])
polygon(7, [th(21), th(24)])
polygon(8, [ht(24), th(19), ht(14), th(10), th(11), th(25)])
```

Figure 2. The process of linking chains to form polygons.

discretization errors. Given the coordinates in rational numbers, the coordinates of the intersection point are always rational, since the intersection point between two line segments is the solution to a linear equation with rational coefficients. Hence we can guarantee stability in the numerical computation for geometric intersections in the polygon overlay process.

Since we can preserve numerical accuracy in our calculations, we can also properly identify the special cases of chain intersections including touching and partially overlapping chains. We limit our handling of special cases to only the primitive operations, which in our case is in intersecting edge segments. We check absolute equality instead of setting a tolerance to identify cases of the end point of an edge segment lying exactly on another edge segment, as well as intersection between colinear edge segments. Thus we consider two edge segments not intersecting if they overlap exactly, and we can identify overlapping chains when forming polygons since they have the same incident angle at their beginning and ending nodes.

Coincidental Input Data and Slivers

Although rational arithmetic offers total accuracy, a realistic problem with map data is coincidental input. Two input maps may have data values of the same feature only approximately equal. As a result, the polygon overlay process generates an output map with sliver polygons which have to be removed. Goodchild studied the problem of slivers and established a measure of the number of sliver polygons related to number of edge segments in the coincidental input data (Goodchild 77). We are developing rules to automatically recognize and remove these sliver polygons. We outline our approach below:

Recognizing Slivers A sliver polygon has a small area and has few bounding edges. We identify three kinds of sliver shapes: a rounded polygon, elongated strip, and a crooked strip. Figure 3 illustrates the different kinds of slivers. They can be recognized by their small area, a small minimum diameter, or a small ratio of area to that of its convex hull.

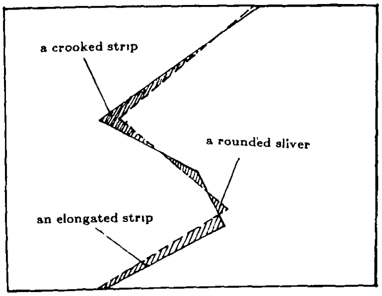


Figure 3. Three kinds of slivers.

<u>*Sliver Removal.*</u> We remove a sliver by coalescing it with one of its neighboring polygons which is not a sliver. This would avoid coalescing slivers to form a non-sliver polygon. A polygon is coalesced to its neighbor by removing the boundary chain in common, and updating the adjacent polygon fields in the remaining boundary chains.

SOME PRELIMINARY RESULTS

The polygon overlay system is implemented using C-Prolog version 1.5, an interpreter written in C. The system runs on SUN 2/170 machine running Unix 4.2 bsd from SUN microsystems release 3.0. Figure 4 shows a test run of two reduced maps with a total of 1720 vertices and 1826 edge segments in 328 chains. The adaptive grid system casted a grid of 116×100 cells onto the scene. 714 pairs of edges were examined and 118 pairs actually intersected. The system uses approximately 11 CPU hours to complete the entire overlay process.

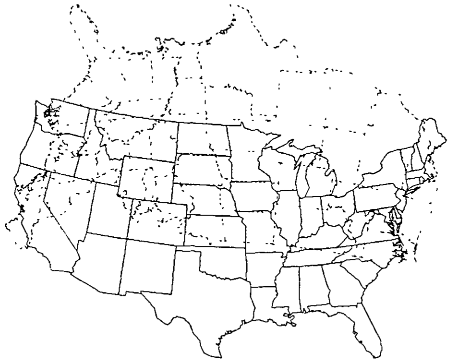


Figure 4. Test example with USA vs USA.

CONCLUSION

We have presented a polygon overlay system developed in Prolog. The system decomposes the complicated process of polygon overlay into various stages. This decomposition allows us to use only simple data structures and mostly local processing operations. Prolog offers a relational database for geometric entities stored as Prolog facts, and a logic programming approach to the encoding of geometry algorithms for data processing. An exact rational arithmetic package enables us to preserve numerical accuracy in calculating intersections. We can then guarantee topological consistency, and properly identify and handle special cases such as touching and overlapping chains. We have also outlined our strategy to remove sliver polygons due to coincidental input data.

ACKNOWLEDGEMENT

This work is supported by the National Science Foundation under grant No. ECS-8351942.

REFERENCES

Chrisman, N.R 1976, "Local versus Global· the Scope of Memory Required For Geographic Information Processing," *Internal Report 76-14*, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Massachusetts.

DeBerry, T 1979, *Polygon Information Overlay System User's Manual*, Environmental Systems Research, Inc., Redlands, California.

Eastman, C.M. and Yessios, C.I. 1972, "An Efficient Algorithm for Finding the Union, Intersection, and Differences of Spatial Domains," Dept of Computer Science, Carnegie-Mellon University.

Ferguson, H R 1973, "Point in Polygon Algorithms," *Urban Data Center Technical Report*, University of Washington, Seattle

Franklin, W R 1972, "ANOTB Routine to Overlay Two Polygons," *Collected Algorithms*, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Massachusetts

Franklin, W.R 1980, "A Linear Time Exact Hidden Surface Algorithm," *ACM Computer Graphics*, Vol 14, No 3, pp.117-123.

Franklin, W.R. 1983, "A Simplified Map Overlay Algorithm," *Proc. Harvard Computer Graphics Conference*, Cambridge, Massachusetts.

Franklin, W R. 1983, "Adaptive Grids for Geometric Operations," *Proc. 6th International Symposium on Automation in Cartography*, Ottawa, Ontario, pp.230-239.

Franklin, W.R. 1984, "Cartographic Errors Symptomatic of Underlying Algebra Problems," *Proc of 1st International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp 190-208

Franklin, W.R., Wu, P Y F., Samaddar, S and Nichols, M 1986, "Prolog and Geometry Projects," *IEEE Computer Graphics & Applications*, Vol 6, No.11, pp 46-55

Freeman, H. and Shapira, R 1975, "Determining the Minimum Area Encasing Rectangle for An Arbitrary Closed Curve," *Communications of ACM*, Vol 18, No 7, pp 409-413.

Gonzalez, J C , Williams, M H and Aitchison, I E 1984, "Evaluation of the Effectiveness of Prolog for a CAD Application," *IEEE Computer Graphics & Applications*, Vol 4, No 3, pp 67-75

Goodchild, M F. 1978, "Statistical Aspects of the Polygon Overlay Problem," *An Advanced Study Symposium on Topological Data Structures and Geographic Information Systems*, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Massachusetts.

Guevara, J A. 1983, *A Framework for the Analysis of Geographic Information System Procedures The Polygon Overlay Problem, Computational Complexity and Polyline Intersection*, Ph.D. Thesis, State University of New York at Buffalo, Buffalo, New York

Little, J.J. and Peucker, T.K. 1979, "A Recursive Procedure for Finding the Intersection of Two Digital Curves," *Computer Graphics and Image Processing,* Vol.10, pp 159-171

Macsyma Group. 1983, *MACSYMA Reference Manual,* Version 10, Vol.II, MIT Press, Cambridge, Massachusetts.

Pereira, F. (ed) 1986, *C-Prolog User's Manual,* Department of Architecture, University of Edinburgh, Edinburgh, U.K

Preparata, F.P. and Shamos, M I. 1985, *Computational Geometry,* Springer-Verlag, New York.

Reed, C.W. 1982, *Map Overlay and Statistical System User's Manual,* Western Energy and Land Use Team, U.S. Fish and Wildlife Services, Fort Collins, Colorado.

Shamos, M.I. and Bentley, J.L 1976, "Optimal Algorithms For Structuring Geographic Data," Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania

Sun Microsystems 1986, *Commands Reference Manual,* Revision G of 17, Part No: 800-1295-02, Sun Microsystems, Inc., Mountain View, California, pp.18-19,82-83.

Sun Microsystems. 1986, *UNIX Interface Reference Manual,* Revision A of 17, Part No: 800-1341-02, Sun Microsystems, Inc , Mountain View, California, pp.270-271.

Swinson, P.S.G 1982, "Logic Programming: A computing Tool for the Architect of the Future," *Computer Aided Design,* Vol.14, No 2, pp 97-104.

Swinson, P.S.G. 1983, "Prolog A Prelude to a New Generation of CAAD," *Computer Aided Design,* Vol 15, No.6, pp 335-343.

Teng, A.T., Joseph, S.A. and Shojaee, A.R. 1986, "Polygon Overlay Processing: A Comparison of Pure Geometric Manipulation and Topological Overlay Processing," *Proc 2nd International Symposium on Spatial Data Handling,* Seattle, pp.102-119.

Tomlinson, R.F , Calkins, H W. and Marble, D.F. 1976, *Computer Handling of Geographical Data,* UNESCO Press, Paris.

White, D. 1978, "A New Method of Polygon Overlay," *Harvard Papers on Geographic Information Systems,* Vol.6, Harvard University, Cambridge, Massachusetts

Wu, P.Y.F. 1986, "Two Arithmetic Packages in Prolog: Infinite Precision Fixed Point and Exact Rational Numbers," *Technical Report IPL-TR-082,* Image Processing Laboratory, Rensselaer Polytechnic Institute, Troy, New York.

COORDINATE TRANSFORMATIONS IN MAP DIGITIZING

Wolfgang Kainz

Research Center Joanneum
Institute for Image Processing and Computer Graphics
Wastiangasse 6, A-8010 Graz, Austria

ABSTRACT

One important task in map digitizing is the conversion of coordinates given in some local device coordinate system to map coordinates. Manual and automatic digitizing devices (digitizing tables, line followers, scanners) send coordinates in inches or metric units to the host. Map data are stored and processed either in rectangular (meters, feet) or geographical units (longitude, latitude). Device coordinates are converted to map coordinates with the help of user defined control points. This conversion may be done with n-parametric polynomial transformations or inverse functions of cartographic map projections. In this paper we investigate both approaches and give some comparative statistics. In the case of polynomial transformations parameter estimation is done with least squares as well as with robust statistical methods.

INTRODUCTION

Manual and automatic digitizing devices produce coordinates that have to be converted to map coordinates. In order to derive results from map data processing the coordinates must be stored in a coordinate system that is suitable for all required tasks. Usually geographical coordinates in degrees of longitude and latitude are chosen, because any cartographic projection may be applied to the data without causing troubles in overlapping zones as it is the case with some projections.

The general problem is to define a transformation between two coordinate systems. In the case of map coordinate data there are two ways of coordinate conversion, using inverse projections and polynomial approximations. Both have already been treated by various authors for digitizing (Fischer 1979) and converting from one projection to another (Brandenberger 1983).

In the case of manual digitizing of paper maps the coordinate conversion is done with the help of control points that are used to compute transformation parameters. The given values of the control point coordinates are used together with the measured values from the digitizing device. It is obvious that these control points have to be measured with utmost care in order to derive useful results. Only one wrong measurement will render unacceptable parameters. In the sequel we shall also investigate methods to decrease the effects of wrongly measured control points.

In this paper we see two procedures applicable to coordinate transformations:

1. Device coordinates --polynomial--> geographical coordinates

2. Device coordinates --polynomial--> intermediate rectangular coordinates (e.g. UTM) --inverse-projection--> geographical coordinates

The first procedure converts device coordinates to longitude, latitude without any intermediate step. In the second case device coordinates are first converted to meters or feet in the projection of the map sheet. Then the inverse projection is used to compute geographical coordinates.

POLYNOMIAL TRANSFORMATIONS

The relationship between digitizing device coordinates (x,y) and geographical map coordinates (long,lat) is expressed by the following formula

$$\begin{aligned} \text{long} &= F1(x,y) \\ \text{lat} &= F2(x,y) \end{aligned} \qquad (1)$$

If we define F1 and F2 to be power series we can write (1) as

$$\begin{aligned} \text{long} &= \sum_{i=0}^{n} \sum_{j=0}^{i} a_{j,i-j} x^j y^{i-j} \\ \text{lat} &= \sum_{i=0}^{n} \sum_{j=0}^{i} b_{j,i-j} x^j y^{i-j} \end{aligned} \qquad (2)$$

The vectors **a** and **b** of the coefficients are estimated as

$$\begin{aligned} \mathbf{a} &= (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{u} \\ \mathbf{b} &= (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{v} \end{aligned} \qquad (3)$$

with **C** being the design matrix composed of the measured values of m control points

$$C = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & \cdots & x_1{}^n & y_1{}^n \\ 1 & x_2 & y_2 & x_2y_2 & \cdots & x_2{}^n & y_2{}^n \\ : & : & : & : & & : & : \\ : & : & : & : & & : & : \\ 1 & x_m & y_m & x_my_m & \cdots & x_m{}^n & y_m{}^n \end{bmatrix}$$

and **u** and **v** the vectors of the given control point coordinates for longitude and latitude

$$\mathbf{u} = (u_1, u_2, \ldots., u_i)^T$$
$$\mathbf{v} = (v_1, v_2, \ldots., v_i)^T$$

The goodness of fit is determined by inspection of the residuals of the given values versus the estimated values of the control point coordinates.

INVERSE PROJECTIONS

When the map projection and all necessary projection parameters are known the inverse projection function can be used to derive longtitude/latitude from given meters or feet. In order to get easting and northing values in meters or feet from digitizing device coordinates a polynomial transformation as in (1), (2) and (3) with control points in meters or feet is used:

$$\begin{aligned} \text{easting} &= F1(x,y) \\ \text{northing} &= F2(x,y) \end{aligned} \qquad (4)$$

These values are then used with the inverse of projection P to compute geographical coordinates.

$$\begin{aligned} \text{long} &= P^{-1}(\text{easting}) \\ \text{lat} &= P^{-1}(\text{northing}) \end{aligned} \qquad (5)$$

ROBUST PARAMETER ESTIMATION

Least squares parameter estimation fail~~e~~s when only one control point is wrong. We have investigated methods of robust parameter estimation using robust and bounded influence regression (Huber 1981, Dutter 1983). The basic principle of robust regression is that we do not minimize the sum of squares of the residuals as with least squares. Instead of the square function robust regression works with functions that give less weight to large residuals, i.e. their first derivative must be bounded.

The program BLINWDR (Dutter 1983) offers linear least squares and three robust options known as "psi bends at c", psi bends at a, b and d" and "psi has the form of sine" which all have bounded first derivatives.

Applying robust parameter estimation in map set-up procedures decreases the effect of inaccurately measured control points (cf. test results below).

TEST RESULTS

For testing the above stated procedures we took one sheet of the Austrian map series 1:50000. The projection of this series is Gauss-Krüger, a variant of the Transverse Mercator projection. The sheet has a Transverse Mercator grid of 2 kilometers. We selected 20 points at grid line intersections. Geographical coordinates were computed using the U.S.G.S. General Cartographic Transformation Package (U.S.G.S. 1982).

Table 1 lists the control points and their coordinates.

Table 1: control point coordinates

	easting	northing	longitude			latitude			digitizer	
	meters	meters	dd	mm	ss.s	dd	mm	ss.s	x-mm	y-mm
1	100,000	5,154,000	14	38	12.0	46	31	05.9	294.525	135.625
2	104,000	5,154,000	14	41	19.6	46	31	03.7	374.250	136.375
3	108,000	5,154,000	14	44	27.2	46	31	01.4	454.175	136.850
4	112,000	5,154,000	14	47	34.8	46	30	59.1	534.425	137.575
5	100,000	5,160,000	14	38	16.6	46	34	20.2	293.450	255.575
6	104,000	5,160,000	14	41	24.4	46	34	18.0	373.325	256.075
7	108,000	5,160,000	14	44	32.2	46	34	15.7	453.100	256.800
8	112,000	5,160,000	14	47	40.0	46	34	13.4	533.375	257.425
9	100,000	5,166,000	14	38	21.3	46	37	34.5	292.425	375.575
10	104,000	5,166,000	14	41	29.3	46	37	32.3	372.275	376.250
11	108,000	5,166,000	14	44	37.3	46	37	30.0	452.100	376.900
12	112,000	5,166,000	14	47	45.2	46	37	27.6	532.350	377.500
13	100,000	5,172,000	14	38	26.0	46	40	48.7	291.325	495.425
14	104,000	5,172,000	14	41	34.1	46	40	46.5	371.225	496.075
15	108,000	5,172,000	14	44	42.3	46	40	44.3	451.150	496.750
16	112,000	5,172,000	14	47	50.5	46	40	41.9	531.400	497.325
17	100,000	5,178,000	14	38	30.6	46	44	03.0	290.000	614.725
18	104,000	5,178,000	14	41	39.0	46	44	00.8	369.825	615.300
19	108,000	5,178,000	14	44	47.4	46	43	58.5	449.700	615.975
20	112,000	5,178,000	14	47	55.7	46	43	56.2	530.050	616.600

Performing parameter estimation for polynomial transformations of degrees 1 and 2 gives 6 and 12 parameters for for both longitude and latitude respectively. Tables 2 shows the results for both procedures described above.

Table 2: residual root mean square in meters

	6 parameters		12 parameters	
	easting	northing	easting	northing
procedure 1	10.42	11.54	4.65	7.21
procedure 2	8.69	11.49	4.65	7.21

For testing the effect of wrong measurements we set the values of point 14 equal to those of point 9, i.e. two different control points are measured at the same location. The test was carried out for procedure 1, table 3 lists the results.

Table 3: root mean square in meters (procedure 1)
robust estmation

	6 parameters		12 parameters	
	easting	northing	easting	northing
least squares	903.95	1364.37	950.35	1425.00
robust estimation	10.86	13.77	5.36	9.04

We have carried out extensive tests with other data sets all leading to the same results as stated above.

CONCLUSION

One can see that with at least 12 parameters (i.e. polynomial degree 2) we can achieve the same result for both procedures. This can be expected as long as the map covers a relatively small area and the projection in use is not too "strange". For very small scale maps procedure 2 seems to be more suitable. It turns out that propably the best way for map digitizing is first to gather coordinates in some local device coordinate system and then apply transformations and/or projections to compute coordinates for a desired coordinate system.

Robust parameter estimation gives little weight to erroneous measurements thus yielding good and acceptable results even if some measurements are wrong.

REFERENCES

Brandenberger, Ch.G. 1985, Koordinatentransformation für digitale kartographische Daten mit Lagrange- und Spline-Interpolation, Dissertation, Institut für Kartographie, ETH Zürich.

Dutter, R. 1983, Computer Program BLINWDR for Robust and Bounded Influence Regression, Research Report, Institute for Statistics, Technical University Vienna, Austria.

Fischer, E.U. 1979, Zur Transformation digitaler kartographischer Daten mit Potenzreihen, Nachrichten aus dem Karten- und Vermessungswesen, Heft Nr. 79, pp. 23-42.

Huber, P.J. 1981, Robust Statistics, Wiley, New York.

U.S.G.S 1982, Software Documentation for GCTP, U.S. Geological Survey, National Mapping Division, Reston.

A SPATIAL DECISION SUPPORT SYSTEM FOR LOCATIONAL PLANNING: DESIGN, IMPLEMENTATION AND OPERATION

Paul J. Densham
Department of Geography
University of Iowa, Iowa City, IA 52242
BLAPPIPD@UIAMVS.BITNET

Marc P. Armstrong
Departments of Geography and Computer Science
University of Iowa, Iowa City, IA 52242
BLAMMGPD@UIAMVS.BITNET

ABSTRACT

In addition to their archival and display functions, spatial information systems have often incorporated an analytical component. Often, this capability has not provided decision-makers with the degree of modelling flexibility and support that they require. Spatial decision support systems are designed to assist decision-making by fully integrating analytical, display and retrieval capabilities; in this paper we describe the development of such a system for complex locational planning problems.

INTRODUCTION

Spatial Decision Support Systems (SDSS) are designed to provide decision-makers with a flexible and responsive problem-solving tool. In this research, an SDSS generator (Sprague, 1980) is designed to help decision-makers find solutions to complex locational planning problems. These problems are combinatorially complex because one or more locations must be selected, for a set of facilities, subject to a variety of constraints. In addition, the set of constraints often cannot be represented mathematically; for example, this occurs when they are political in nature, or are poorly defined. Consequently, traditional optimizing analysis cannot be applied <u>alone</u> to derive an optimal solution. Mathematical models <u>are</u>, therefore, used as <u>part</u> of a solution process, in which a series of feasible solutions is produced and evaluated against a set of defensible decision criteria to yield an optimal solution (Densham and Rushton, 1986).

Typically, an SDSS contains a spatial information system integrated with a modelling system. More specifically, it includes a geo-referenced database and modules, to provide analytical, display and reporting capabilities. This paper describes the development of a microcomputer-based SDSS; it is organized in four major sections - design, implementation, operation, and prospects.

DESIGN

SDSS and Decision-Making

An SDSS integrates analytical techniques with the expertise of decision-makers, placing the emphasis of the approach on making effective decisions (Keen and Scott-Morton, 1978; Alter, 1980). Figure 1 gives a schematic representation of the components of a microcomputer-based SDSS generator for spatial analysis (Armstrong, Densham and Rushton, 1986). It is designed to support a decision process that redefines the concept of optimality used in analysis. Keen (1977) uses Simon's three-stage model of decision-making (intelligence, design and choice) to show that traditional optimizing analysis emphasizes the choice stage, because it focuses on the optimal nature of the solution. Consequently, in traditional analyses, optimality has been defined as a characteristic of the solution. A contrasting approach to decision-making is to generate and investigate alternative solutions (Hopkins, 1984), emphasizing the stages of intelligence and design in the analysis. In this approach optimality becomes a characteristic of the whole decision process, encompassing all aspects of the problem, including those that cannot be represented in the objective function. It is this latter concept of optimality that underlies the SDSS decision process.

As Keen (1977) notes, this form of solution procedure is generally iterative. Each alternative is presented to the system user as formatted reports, maps and graphs created by the reporting and graphics modules. The alternatives are then evaluated by decision-makers, using their expert knowledge, against a set of criteria consisting of those initially thought to be important, and those not previously considered that are "uncovered" while examining alternatives. Decision-makers can accept an alternative as a satisfactory final solution or they can attempt to improve the solution by using feedback loops to modify the parameters of a model, or to specify a new one. Thus, using their understanding of the problem, decision-makers produce solutions that are optimal with respect to the dimensions and aspects that they consider to be important.

Database Requirements of Spatial Analysis

Spatial analysts use many modelling techniques. Among their repertoire is location-allocation modelling, which enables an analyst to locate one or more facilities, and to allocate demand to each facility, by optimizing the value of an objective function. This technique uses locational, topological, and thematic data which, in concert, provide the capability to capture a rich representation of the geography of a given area. Analysts require a general spatial data structure that can store and manipulate these data at a variety of spatial scales and degrees of attribute resolution (Anderson, 1978; Elmes and Harris, 1986). To support analyses, this data structure should enable specification and analysis of shapes, distances and directions, and must make comprehensive display of the data possible. Also, the data structure must easily accommodate variability in topological dimension and precision. Finally, thematic data are generally recorded in both a chronological and a categorical manner. Thus, in addition, the data structure should enable simple retrieval and manipulation of these data by time period, by category, and by both indices.

The basic element of a location-allocation model in discrete space is a graph, or network, consisting of a set of nodes and links. The nodes represent demand points, which are geo-referenced using an absolute or a relative coordinate system. Links depict transportation corridors between two nodes. The graph may be directed, signifying that only restricted movement along one or more links is possible. The friction of travel through space is represented by a "distance" value for each link - the unit is a measure of time or distance.

IMPLEMENTATION

The Database

There are many data models which can be considered for use in an SDSS; these include the rectangular, network, hierarchical, relational and extended network models. The extended network model (Bonczek, Holsapple and Whinston, 1976) has been selected because it will efficiently support the set of general capabilities described above. Also, Bonczek, Holsapple and Whinston (1981) have shown that the extended network model is a good foundation for general Decision Support Systems (DSS). The system set provides a powerful construct for directly accessing data in various locations in a database. This reduces both software development time and access times for data retrieval, because data can be accessed directly rather than by traversing intermediate records. It also enables the designer to produce a database that appears to be very close to the user-view of the data structure, yielding both flexibility and ease of use.

The database has been produced using Microsoft Pascal (Version 3.31) and MDBS Incorporated's MDBS III on an IBM PC/XT and a Leading Edge Model D. Figure 2 shows the logical structure of the implemented database, and illustrates the tripartite classification of data. This equates locational data with point and chain spatial primitives; and topological data with attribute-bearing entities such as the node, line and cell. Similarly, thematic data are represented by attributes of the topological entities stratified in a temporal data sequence. States, cities, chains and points are each defined to be system sets.

The database must be able to represent both uni-directional and bi-directional links in the network, and must record which links fall into each category. There are two 1:N (one-to-many) relationships between points and nodes enabling a point to own chains in two different sets, which are built when the user enters a chain. These sets are used to generate the appropriate data structure for standard or directed graphs.

The relationships between the topological and locational data are designed to support many levels of spatial precision. In a location-allocation framework, for example, the scale of analysis may be intra-urban or inter-urban. Each city can own one or more points and one or more chains; consequently, a city can either have an areal extent, with a boundary comprised of chains, or it can be represented by a single point. This allows both intra-urban and inter-urban networks to be established, and analysis to be undertaken at a spatial scale commensurate with both the available data and the objectives of the analysis.

The thematic data consist of six different record types; those owned by states and cities are essentially identical except for their spatial scale - both are sorted by date and contain data on variables such as population size. The third record contains the name and type of linear features represented by chains; whereas the fourth stores the name and type of point features.

The fifth type of record is owned by the chain, and consists of four fields that contain distance and distance-related data. The first field is the Euclidean distance between the endpoints of each chain, the "from" and "to" nodes. The next field is the sum of the Euclidean distances calculated between all the points defining a chain. These two values are calculated when the user enters the chain into the database. The third field contains a distance or time value specified by the user. The final field is the fractal dimension of the chain, which could be used in conjunction with low resolution data to provide realistic graphic displays (Dutton, 1981).

A node owns one or more of the sixth type of record, which consists of data required by the location-allocation routines. Five items of data are required for each of the nodes:

1) The "set" number is the identifier for each of the multiple node records.

2) The unique node identifier.

3) The demand for the service (provided by a facility, or facilities) is aggregated over space to the proximal node on the network; it is termed the "weight" of the node (Goodchild and Noronha, 1983).

4) The fourth value shows if there is a facility at a node that cannot be relocated, constraining the location-allocation heuristic to preserve existing facilities in the solution.

5) The "candidacy" of a node describes whether or not that location is suitable for a facility.

Interfacing the Database and Analytical Modules

The analytical module contains an extended version of the PLACE suite (Goodchild and Noronha, 1983) of location-allocation programs, recoded from BASIC into Pascal; it is linked to the database using a software interface. The

BASIC version of the PLACE suite requires that data are stored in files. These variables include the node weight and candidacy, the "from" and "to" nodes of each link, and its associated distance value. The node identifiers of any fixed facilities in the network are entered interactively at run-time. In contrast, the SDSS interface between the analytical module and the database retrieves data and passes it directly to the arrays used by the location-allocation heuristic.

New features permit easier and more flexible model building than can be achieved with the PLACE suite. The "set" number, in each node record, identifies each of the multiple records linked to a node, permitting the compilation and storage of different data sets in the same database. Consequently, many different analyses can be carried out easily on the same network. The interface also enables the user to specify which of the three distance values is to be retrieved and passed to the analytical module. A corollary of the calculation of link distances by the database is that the SPA5 algorithm in the PLACE suite becomes redundant, and is discarded. The interface can produce data files for the BASIC version of the PLACE suite, maintaining backward compatibility and transportability of data sets. Finally, the interface employs a number of checks for data inconsistencies.

Graphics

The display module is coded in Microsoft FORTRAN (Version 3.31) using Lifeboat Associates' HALO graphics package (Version 2.01). An interface to the database and location-allocation modules, written in Pascal, is being refined. Its function is to pass data from the database, and the results of an analysis, to the graphics module. In concert with commands from the user interface, this module produces displays of the solutions for interpretation by the users. Maps show where facilities have been located on the network, and graphs enable the user to evaluate statistical descriptions of each solution (Schilling, McGarity and ReVelle, 1982). In addition, base maps of the study area can be produced directly from the database. Various degrees of spatial abstraction can be represented on the maps, enabling the user to overlay a "landscape" to provide a frame of reference. The routines in the graphics module are being written in a modular fashion in order to facilitate migration to the GKS and virtual device interface environment.

OPERATION

The SDSS is built from several modules, which are functionally and logically distinct, corresponding to Sprague's (1980) general framework for the development of decision support systems. This modular framework has several advantages to it; the first is that a modular system is easily produced from a synthesis of existing, often commercially available, software. In addition, such a system is easily extended. New modules can be integrated with a minimum of re-coding, and maintenance of both the individual modules and the entire system is facilitated. However, a modular system can be very hard to use if a variety of existing interfaces and command syntaxes are incorporated in one system. Consequently, the operation of this SDSS is designed to be "seamless." By this it is meant that the system will appear to function as an integrated unit under an overarching, standardized interface. This structure will make the modularity of the underlying software components transparent to the user.

PROSPECTS

The PLACE suite of programs provides an excellent stand-alone location-allocation package. The integration of the suite into the SDSS, however, provides an opportunity to exploit the capabilities of other system modules; consequently, a number of extensions are planned that will enhance the flexibility of the location-allocation module over that of the PLACE suite. The first is to let the user define "templates" that will designate which links and nodes are to be dropped from, or added to, the network for analyses. Templates will enable the user to analyze partial networks derived from the main network in the database. This capability will enable a user to study each network at varying degrees of detail; and, in concert with the various spatial scales that are supported by the database, they will be able to carry out a wide range of analyses on a given database. The second extension is to enhance the reporting capabilities of the SDSS over those of the EVAL program in the PLACE suite. This change will both increase the information presented to the user and make the reports complementary to the graphical output.

The SDSS is designed to be used by a broad spectrum of users. Consequently, ease of use is an important consideration and three expert systems will be added to the SDSS to act as local or global expert controllers. A pair of local expert controllers will oversee the operation of the modelling and the graphical and reporting modules; similarly, a rule-based user interface will act as a global expert controller for the SDSS. The graphical and reporting expert controller is being developed at present.

Many applications of cartographic expert systems perform only a subset of the tasks involved in virtual map production. Based on reasoning techniques, they also have a large overhead in terms of code and computation speed that render them inappropriate for use in a microcomputer-based SDSS. However, part of the map production process can be viewed as a problem of pattern recognition, rather than reasoning; one of matching the attributes of the data to be displayed with those of various map types.

Amongst many artificial intelligence techniques for pattern recognition are Holland Classifiers (Holland, 1975, 1986), which are based on bit-mapping techniques that are fast when compared with many AI techniques. The dichotomy between a knowledge base and inference engine is maintained when the classifier is implemented in a procedural language. The classifier uses a similarity index to determine whether or not a bit string, describing the attributes of the data, matches the string associated with a particular form of map. The use of multi-letter alphabets, rather than a binary one, make it possible to define importance classes for the attributes and carry out fuzzy matching (Leung, 1983) of the bit strings.

In the system under development, the attributes will be provided from both the user and a pre-processor in the graphics interface. The user will set the values of some attributes such as the color and type of symbolization. This can be done using global variables, with the system having a set of cartographically sound default values. The pre-processor will calculate values such as the degree of spatial abstraction, and the scale of the map, from the results of the analysis and the data in the database. The classifier will then match the attributes with the bit strings describing the various forms of map, and produce the appropriate one using the results from the analysis and the locational, topological and thematic data stored in the database.

SUMMARY

A spatial decision support system for locational planning has been implemented on an IBM PC/XT. It uses location-allocation modelling heuristics, in concert with database graphics and reporting modules, to provide the user with a tool to support an iterative solution process. The integration of global and local expert controllers is being investigated, beginning with one to oversee virtual map production.

REFERENCES

Alter, S.L., 1980, Decision Support Systems: Current Practice and Continuing Challenges, Addison-Wesley, Reading

Anderson, K.E., 1978, Spatial Analysis in a Data Base Environment: Proceedings, First International Advanced Study Symposium on Topological Data Structures for GIS, Dutton, G., ed., Vol. 2

Armstrong, M.P., Densham, P.J., and Rushton, G., 1986, Architecture for a Microcomputer-Based Spatial Decision Support System: Proceedings, Second International Symposium on Spatial Data Handling, Seattle, pp. 120-131

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B., 1976, Extensions and Corrections for the CODASYL Approach to Data Base Management: Information Systems, Vol. 2, pp. 71-77

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B., 1981, Foundations of Decision Support Systems, New York, Academic Press

Densham, P.J., and Rushton, G., 1986, Decision Support Systems for Locational Planning: Behaviour Modelling Approaches in Geography and Planning, R. Golledge and

H. Timmermans, eds., Croom Helm, London

Dutton, G.H., 1981, Fractal Enhancement of Cartographic Line Detail: The American Cartographer, Vol. 8, pp. 23-40

Elmes, G.A., and Harris, T.M., 1986, Hierarchical Data Structures and Regional Optimization: An Application to the Sussex Land-Use Inventory: Proceedings, Second International Symposium on Spatial Data Handling, pp. 289-305, Seattle

Goodchild, M., and Noronha, V., 1983, Location-Allocation for Small Computers, Monograph 8, Department of Geography, The University of Iowa, Iowa City, IA

Holland, J.H., 1975, Adaptation in Natural and Artificial Systems, Ann Arbor, University of Michigan Press

Holland, J.H., 1986, Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems: Machine Learning II, Michalski, R.S., Carbonelli, J.G., and Mitchell, T.M., eds., Los Altos, Morgan Kaufmann

Hopkins, L.D., 1984, Evaluation of Methods for Exploring Ill-Defined Problems: Environment and Planning B: Planning and Design, Vol. 11, pp. 339-348

Keen, P.G.W., 1977, The Evolving Concept of Optimality: Multiple Criteria Decision Making, Starr, M.K., and Zeleny, M., eds., New York, North-Holland

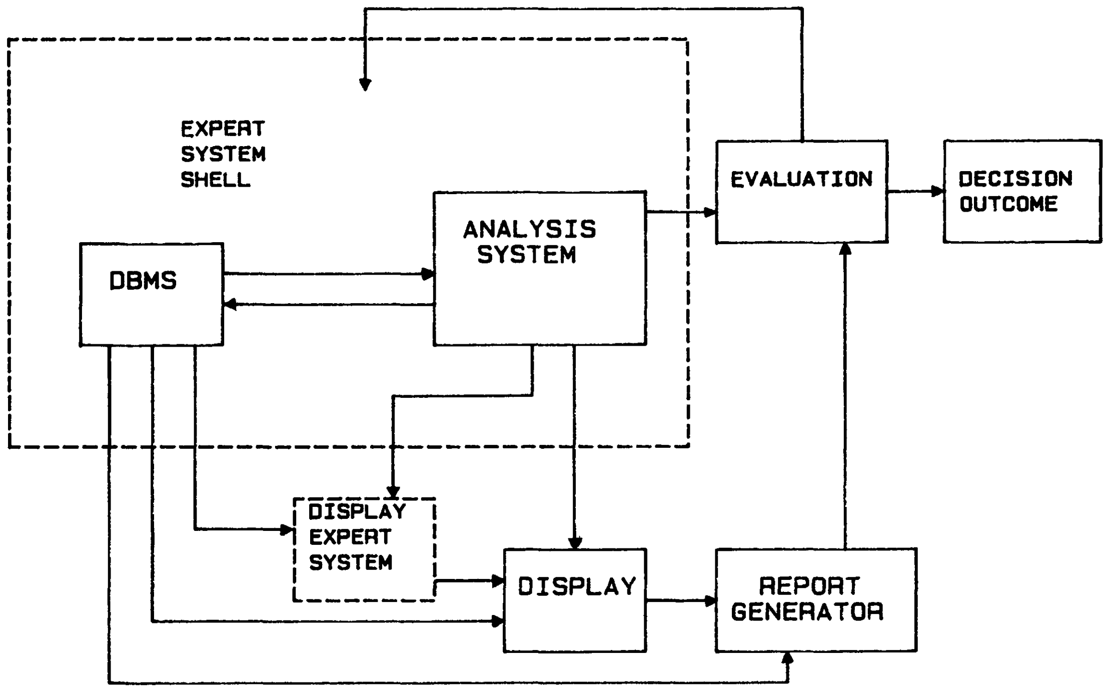
Keen, P.G.W., and Scott-Morton, M.S., 1978, Decision Support Systems: An Organizational Perspective, New York, Addison-Wesley

Leung, Y., 1983, Fuzzy Sets Approach to Spatial Analysis and Planning - A Nontechnical Evaluation: Geografiska Annaler, Series B, pp. 65-73

Schilling, D.A., McGarity, A., and ReVelle, C., 1982, Hidden Attributes and the Display of Information in Multiobjective Analysis: Management Science, Vol. 28, pp. 236-242

Sprague, R., 1980, A Framework for the Development of Decision Support Systems: Management Information Sciences Quarterly, Vol. 4, pp. 1-26

FIGURE 1: SOFTWARE COMPONENTS FOR SDSS



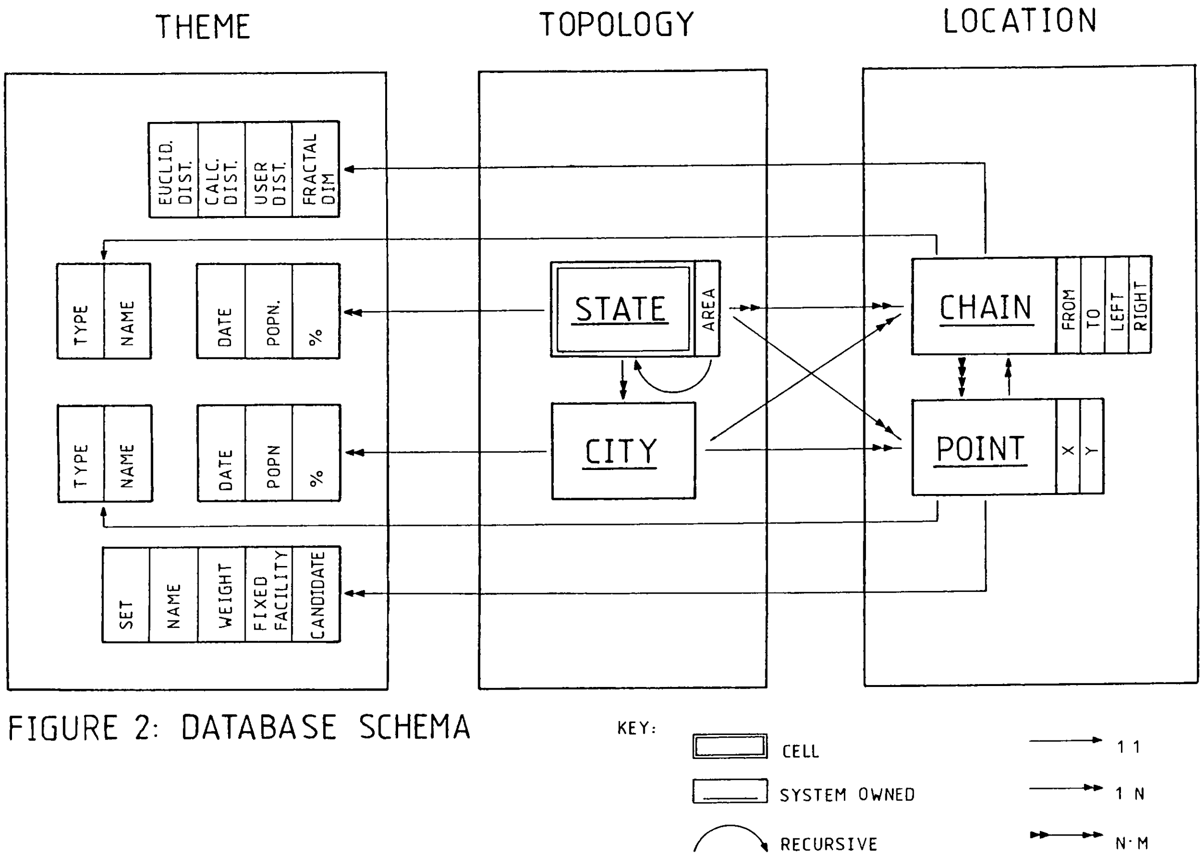


FIGURE 2: DATABASE SCHEMA

Realistic Flow Analysis Using a Simple Network Model

William H. Moreland
Anthony E. Lupien

Environmental Systems Research Institute
380 New York Street
Redlands, California 92373

ABSTRACT

Simple topological data structures consisting of links and nodes have often been used for modeling flows through a network because of their processing efficiency and ease of implementation. These representations of networks are typically assumed to be flat and metrical; links have the same cost of travel in either direction and no costs are associated with traversing nodes. This paper describes three additions to the common model that retain its advantages while offering a capability for much more realistic analysis of network flows.

The first enhancement of the simple network model is the assignment of separate costs for traversing a link in either direction. This simple modification allows consideration of physical constraints such as slope and temporal contraints such as traffic. Second, costs are associated with each link-to-link turn. This allows consideration of the delays experienced at network nodes from congestion and controls, and permits impossible turns (e.g. from an overpass) to be removed from paths of flow. Finally, it is shown through examples that integration of an enhanced network model with a relational database management system simplifies the assignment of appropriate travel costs enough that a single network can support realistic analyses that consider the unique characteristics of various types of links and nodes and flows across them in various congestion and control scenarios.

Realistic Flow Analysis Using a Simple Network Model

A network is a weighted connected graph. That is to say a network is a series of connected links and nodes, where each link has associated with it data identifying its characteristics for allocation. Figure 1 shows the components making up a simple network. A prime example of a simple network is a street map (figure 2), the streets can be thought of as links and the intersections as nodes. Because street layouts are easily represented as networks, there is a lot of interest in optimizing flow through them. With that in mind, there has been a lot of work in devolving algorithms for modeling flow through networks. For the ever present reasons of speed and memory, simple networks are better suited for computer modeling of flow. Simple networks characteristically have only one link between any two nodes and assign a single impedance.

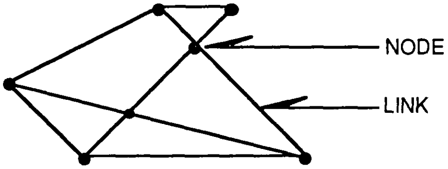


Figure 1.

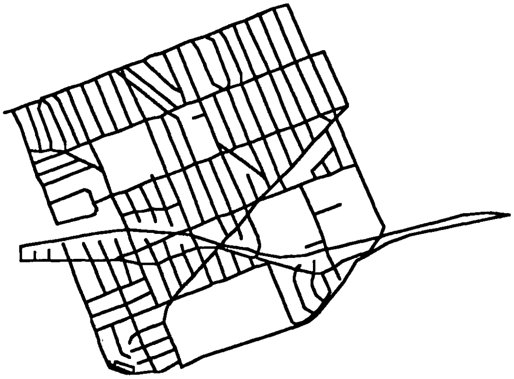


Figure 2.

Realistic Flow Analysis Using a Simple Network Model

Allocation through a simple network model involves the growing out from one or more centers across the links. The method utilized at ESRI is a variation of Loubal's [1] extension of the well-known Moore [2] or Dijkstra' [3] Minimal Path Algorithm, but involves a heap structure to manage the selection of links. A heap is an ever sorted list, designed so that the next item off is always the smallest of all of the items currently kept within it. The heap is made up of link number and impedance pairs, where the impedance is the controling factor for the heap.

We are going to use the one center case within this paper; but keep in mind that the multiple center case uses the same heap structure, but each link keeps tract of which center it is allocated to. The one center case is illustraded by allocating traffic flow from or to a central point, while the multiple case is illustrated by allocating students to all of the schools within a city.

The heap starts out with an nonexistence link with a zero impedance whose next node to traverse is the center. Now starts the allocation process of taking the next link, called **L**, and its impedance, called **I**, from the heap. Once off the heap, all possible links directly reached from **L** are located together with their respective impedances. Each possible link is assigned a new impedance, which is the sum of its own impedance and **I**. The link is then placed back onto the heap. The allocation stops when the heap is finally empty. This method insures that each link is reached by the optimum path based solely upon impedance.

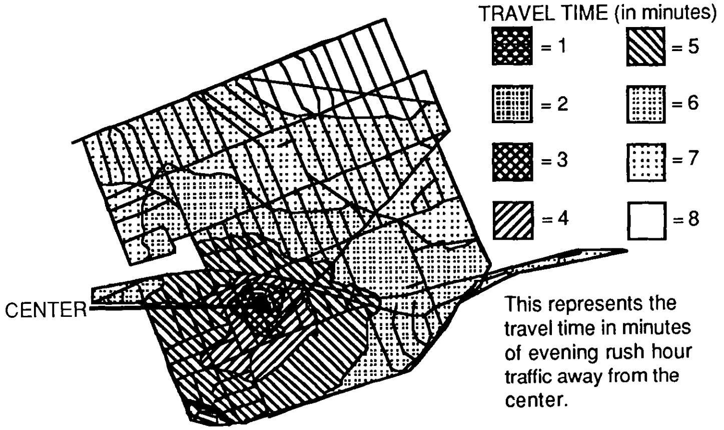
The simplicity of a non-directional network is due in part to the number of impedances that each link must maintain and incorporate in the model, and the fact that there is only one link connecting any two nodes. By always having only one impedance at the time of allocation assigned to each link, the framework for a simple network is maintained. The single impedance model affords easy implementation on the computer, but for the network that allows the same flow in both directions across a link, real-life traffic conditions cannot be easily handled if at all.

The directional network which allows different flow in either direction across a link offers a solution to the modeling of most traffic conditions; however, it poses still more problems on the handling of the differing impedances that each link can have. By associating the network with a relational database management system, the problem is reduced in magnitude to a level that allows for easy implementation.

The relational database allows the impedances assigned to each link to be selected, changed, or easily modified before each allocation. The model can now support multiple impedances that reflect the various traffic conditions associated with different times of the day; consequently the same network model can be used without modification to perform mulitple allocations of differing traffic scenarios.

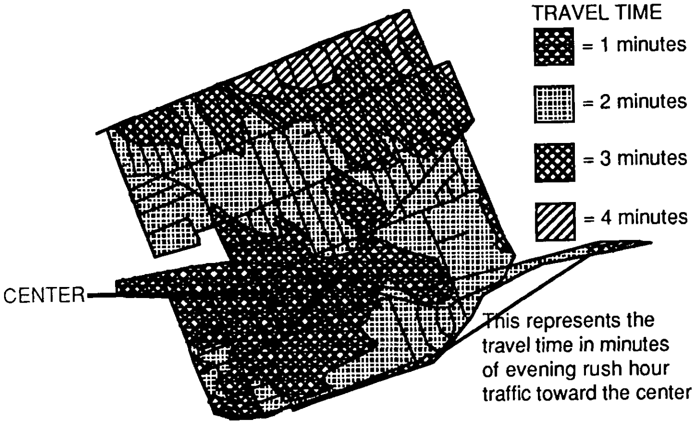
Realistic Flow Analysis Using a Simple Network Model

The allocation process itself assigns direction of travel to each link once it is added to the tree structure, and therefore the correct impedance for the link, based solely upon the direction of travel, can be assigned. In this way, the single impedance network model is still maintained. With the addition of directional impedance, the model now more closely reflects the actual flow of traffic through the network. The model can now handle the real-life conditions; while maintaining the speed normally associated with allocating flow through a simple network. Figure 3 displays a model of travel time during rush hour traffic conditions; while figure 4 displays the same model, but with traffic flowing toward the center instead of away from the center.



This represents the travel time in minutes of evening rush hour traffic away from the center.

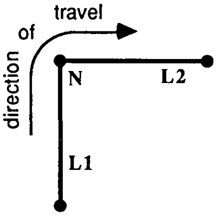
Figure 3.



This represents the travel time in minutes of evening rush hour traffic toward the center

Figure 4.

Since the model is closely linked to a relational database management system, another impedance can be easily added to the model. This impedance, called turn impedance, controls the rate of travel from one link onto another. This still does not imply that the simple network approach can no longer apply. The single impedance assigned to each link, in addition to its appropriate directional impedance, will also reflect the impedance of the turn needed to travel onto the link.

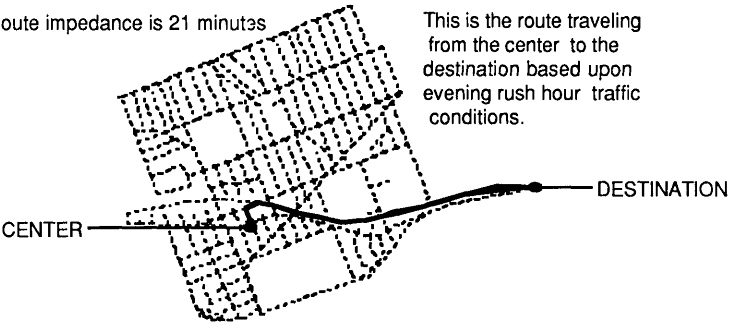


The turn impedance is the rate of travel from link one (L1) to link two (L2) across node (N)

Figure 5.

The turn impedance is the cost of traveling from one link across a node onto another link (figure 5). These impedances are maintained in the same relational database, offering easy selection and modification before allocation. With the addition of turn impedance, the model now closely reflects the actual flow of traffic through the network. The model can now handle restricted flow through the nodes without having to stop all flow. This incorporates such network considerations as: Overpasses, Underpasses, Highway on and off ramps, Turning from or onto one-way streets, U-turns, Left-hand versus right-hand turns, etc...

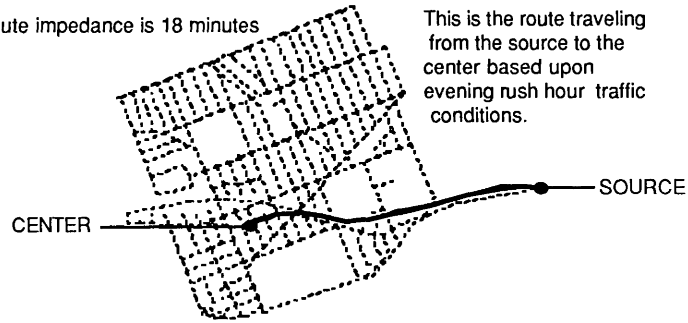
Figure 6 is an example of using turn and directional impedances to perform route evaluation. In figures 6, 7, 8, and 9; left hand turns were assigned an impedance of 5 minutes, right hand turns an impedance of 10 seconds, no turns an impedance of 30 seconds, and U-turns were disabled. Figure 9 brings together all of the impedances, in order to solve an real-life traffic scenario. For this example figure 8 shows the route without the addition of an accident, which completely disables the intersection. The difference of impedances between figures 6 and 7, illustrates how directional impedances alter the allocation process based upon the direction of travel along the links.



Route impedance is 21 minut3s

This is the route traveling from the center to the destination based upon evening rush hour traffic conditions.

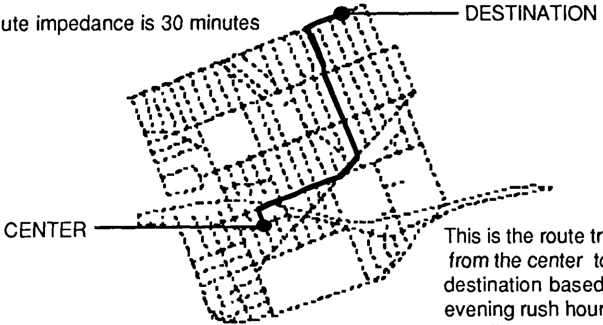
Figure 6.



Route impedance is 18 minutes

This is the route traveling from the source to the center based upon evening rush hour traffic conditions.

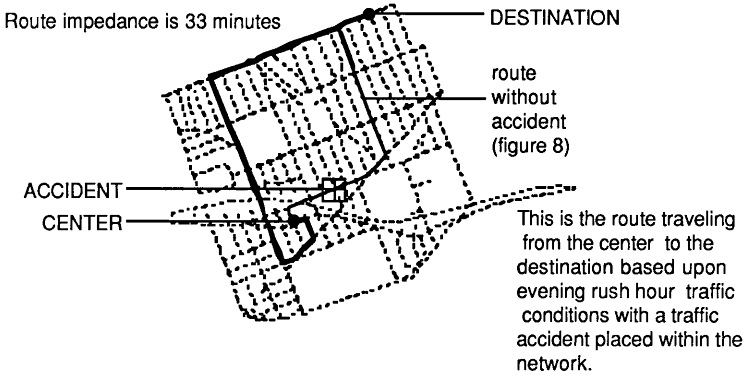
Figure 7.



Route impedance is 30 minutes

This is the route traveling from the center to the destination based upon evening rush hour traffic conditions.

Figure 8.


Route impedance is 33 minutes
DESTINATION
route without accident (figure 8)
ACCIDENT
CENTER

This is the route traveling from the center to the destination based upon evening rush hour traffic conditions with a traffic accident placed within the network.

Figure 9.

The flow through the network can now be modeled in such a way that it closely follows real-life traffic conditions. The speed of the allocation process is still preserved by virtue of the fact that the heap structure must only maintain the two items of link number and cumulative impedance.

The methods of allocation brought forth in this paper, as illustrated by traffic flow through a street network, may be applied equally well to most flow analysis modeling problems.

REFERENCES:

1) P. S. Loubal, A Procedure for Allocating Resources Over a Network.
2) E. F. Moore, The Shortest Path Through a Maze,
Proc. Int. Symp. on the Theory of Switching,
Harvard University, Cambridge, Massachusetts, 1-3 (1963).
3) E. W. Dijkstra', A Note on Two Problems in Connection with Graphs,
Numerische Mathematik, 1, 269-271 (1959).

A GEOGRAPHIC INFORMATION SYSTEM UTILIZING THE TRIANGULATED IRREGULAR NETWORK AS A BASIS FOR HYDROLOGIC MODELING

Andrew T. Silfer[1], Gerald J. Kinn[2] and James M. Hassett[3]

[1] Camp Dresser & McKee Inc.
Raritan Plaza One
Raritan Center
Edison, New Jersey 08818

[2] TASC
100 Walkers Drive
Reading, Mass. 01867

[3] SUNY College of Environmental Science and Forestry
312 Bray Hall
Syracuse, New York 13210

ABSTRACT

The TINFLOW system is a PC-based Geographic Information System (GIS) that utilizes the Triangulated Irregular Network (TIN) and associated data structures, together with a deterministic, finite difference approach, to model rainfall-runoff processes via overland flow and interflow. The Triangulated Irregular Network is used to accurately model a watershed as a series of triangular facets. The TIN methodology and data structure allows the user to conveniently store or directly calculate the necessary physical information of the basin required by the hydrologic model. In addition to these parameters, attributes such as soil type or cover type may be specified and stored directly in the data structure. These attributes allow the user to specify, on a facet-by-facet basis, the physical parameters that drive the hydrologic model. This type of analysis is particularly well suited to modeling of urban areas with its alternating areas of pavement and vegetation. It may also offer the capability to predict the results of change within a watershed (for example, a clearcut area's effect on rainfall- runoff, water quality and soil erosion).

TINFLOW SYSTEM DESCRIPTION

TINFLOW is a geographic information system (GIS) written for a personal computer environment in Turbo Pascal. The GIS contains a hydrologic module that can predict a stream's response to storm events. It is useful in predicting flood levels, in maximizing hydroelectric power generation from a given storm, in determining release schedules of reservoir systems to optimize water storage, and in understanding stream characteristics that may influence the design of flood control structures, dams, and habitat improvements.

Over the last fifty years many methods have been developed to predict the outflow hydrograph of a stream. These methods range from the unit hydrograph theory, which is applicable only if measured flow data is available for that stream, to mathematically complex hydrologic models that predict a watershed's outflow hydrograph based on the physical characteristics of the watershed. Examples of these models are the Stormwater Management Model (SWMM) and the Stanford Watershed Model.

The majority of these models discretize the watershed into a small number of subwatersheds. Each subwatershed is then assigned one number, or index, for each physical characteristic of the land such as slope or land cover type. It is logical to assume that the more discretized a watershed, the more accurately physical attributes can be assigned. These attributes are what determine a watershed's response to storm events.

The TINFLOW system also uses a discretized watershed, with physical attributes assigned to each subarea, as a basis for the associated hydrologic model. However, the method used to discretize the watershed is unique. The TINFLOW system employs the Triangulated Irregular Network (TIN) as a digital terrain model to represent the topography of the watershed.

TIN FORMAT VS. MATRIX FORMAT

The primary method for the production and distribution of digital terrain models is the matrix format, a raster of equally spaced elevation posts. The question then arises, what value does the TIN format offer over a matrix format for this particular application? The answer becomes apparent when the modular character of the solution is examined. It is desirable for software solutions to be very modular. This characteristic allows easy modification and makes future development more cost-effective.

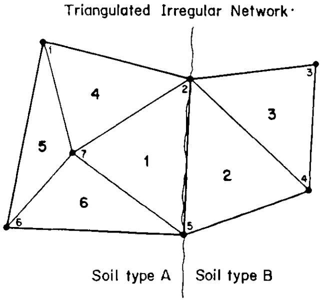
The elemental unit for the terrain matrix is a point. A data point, however, does not allow meaningful analysis of terrain. The elemental cluster in a matrix would consist of four points, or a two-by-two cell. This represents a complex surface because it can be curved between elevation posts. Similiarly, depending on the interpolation schemes, the boundary conditions of this elemental cluster may be quite complex.

The TIN model consists of a collection of triangular planes joined at their boundaries. The spacing and shape of the triangles is determined by the terrain and by the desired degree of fit. Even though the elemental information is a point, the elemental cluster is always a triangle. The interpolation scheme is assumed to be linear, therefore, the few boundary conditions are straight forward to specify.

The development of a hydrologic model for a plane surface is trivial. If a model can be developed to handle the set of all possible boundary conditions to that plane, then the complete solution for a TIN terrain model can be specified. Furthermore, if the software is developed in this way, the individual developing the hydrologic model needs only to be concerned with one routine that describes the flow over a plane. This highly normalized approach lends itself to a modular software solution.

SYSTEM COMPONENTS

The TINFLOW GIS's data requirements are created from two initial data files: a topology file and a node coordinate file. The TIN topology file contains topologic and attribute data for each facet. An example of the structure used in this study is illustrated in figure 1. The second data file contains node coordinate information. In this file each node's location—latitude, longitude, and elevation—are stored. Two preprocessing algorithms, PREPRO and CHECKR are used to perform calculations and place the data into the structure required by the hydrologic model.



TIN Data structure: OPPIJ - Facet opposite IJ node link

TIN facet #	I node #	OPP IJ	J node #	OPP JK	K node #	OPP KI	Soil Attribute
1	2	2	5	6	7	4	A
2	2	3	4		5	1	B
3	3		4	2	2		B
4	1		2	1	7	5	A
5	1	4	7	6	6		A
6	7	1	5		6	5	A

Blank fields denote the edge of the TIN.

Figure 1—An example of a TIN data structure

The hydrologic model is used to simulate the watershed's runoff hydrograph. A postprocessing menu system that immediately follows the model allows the user to graphically view the results of each hydrologic simulation.

ALGORITHM DESCRIPTIONS

PREPRO Algorithm

The PREPRO algorithm performs several important operations.

- It performs a relational join of the topologic and node coordinate files to form a third hybrid data structure.
- It calculates physical parameters of each TIN facet such as slope, area, and aspect, from the TIN geometry.
- It determines how water should be routed across the surface of each TIN facet.

One of the most difficult problems involved in developing a dual digital terrain and hydrologic model is the linking of individual TIN facets hydrologically. During storm events, one of the paths water follows to a stream is over the land's surface; therefore , it is important to understand how water flows across an individual TIN facet surface, and from one facet's surface to another.

All possible flow cases were examined during development of the mode. It was determined that there were two basic conditions that can occur:

- Flow enters two sides of a facet and exits from one side.
- Flow enters one side of a facet and exits from two sides.

To determine which flow case exists for each facet in the watershed, the PREPRO algorithm first calculates the equation of the vector normal to the facet's surface, as shown in Figure 2.

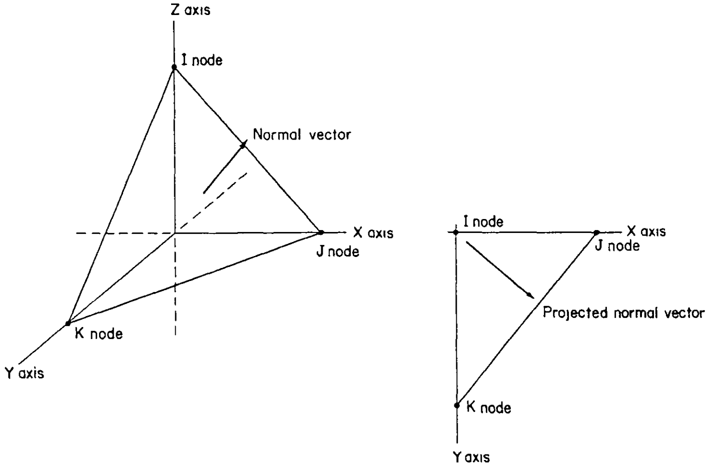


Figure 2—Schematic view of a facet showing the normal vector

This vector, when projected onto the x-y plane, defines the line of maximum slope and thus the direction that water would flow across the facet's surface. The x and y components of the projected vector are used to find the aspect of the fall line or the direction of flow for that facet.

One of the major challenges involved in the design and implementation of this algorithm was devising decision rules that would distinguish between the expected flow cases. Each facet's geometric orientation is evaluated and one of two fundamental flow cases is assigned. Case one has flow entering two sides of a facet and exiting one side, thus called In-In-Out or I-I-O. Case two has flow entering one side and exiting two, and is called I-O-O. Because the chance of encountering a level facet in a natural watershed is very small, this situation was ignored. Cases involving flow entering or exiting three sides of a facet are viewed as null cases in a natural watershed, since sewers and manholes do not exist.

A decision rule was developed to distinguish between the I-O-O and I-I-O cases. During creation of the TIN facets, the I node label is assigned to the one of the facet's three nodes with the highest elevation. From the I node, one moves clockwise around the facet, with the first node encountered being assigned the J node label, and the second node being assigned the K node label. The decision rule involves examining the aspects of the IJ and IK vectors. Conceptually, the facet is placed on a set of cartesian axes, with the I node positioned at the origin. Next, the aspects of the IJ vector, the IK vector, and the facet's slope aspect are drawn on the axes. Then, beginning at the IJ vector, one moves in a clockwise direction across the facet's surface. If the facet's slope aspect is encountered before the IK vector then the facet is oriented as an In-Out-Out case, as illustrated in Figure 3. If this decision rule is not met, the case is ruled an In-In-Out scenario.

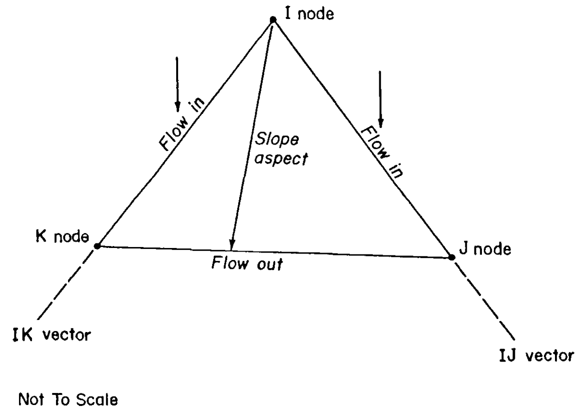


Figure 3—Schematic of typical In-In-Out flow case

To represent the flow of water across a discretized surface, it is important to route the flow accurately between adjacent TIN facets. PREPRO uses the flow case information to calculate the percentage of flow exiting from the current facet of interest to the downhill facets. This problem is simplified for the In-In-Out case since flow exits totally to one TIN facet.

For the In-Out-Out case, the slope aspect line, which defines the direction of runoff for that facet, is placed through the node with the lowest elevation. PREPRO then solves for the coordinates of the intersection point of the slope aspect and the opposite node link. The area to the left of the fall line, divided by the facet's total area, represents the percentage of that facets outflow draining to the facet opposite this JK node link. Likewise, the area on the right of the fall line divided by the total facet area represents the percentage of flow exiting to the facet opposite the KI node link, as shown in Figure 4. These neighbor relationships are easily determined from the TIN topology. Adopting the convention that inflow is positive and outflow is negative, the database is then updated with the outflow percentages so that they may be accessed later by the hydrologic model.

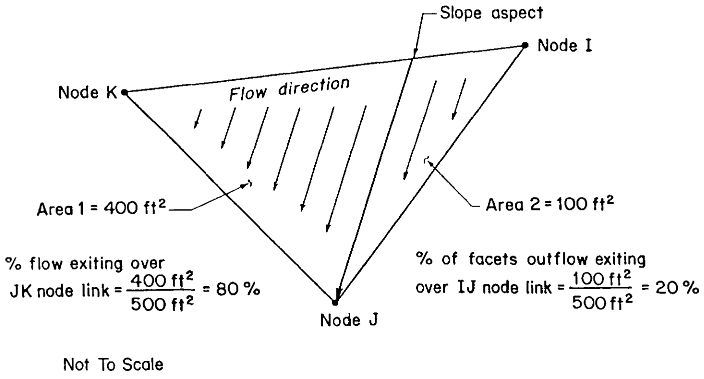

Slope aspect
Node I
Node K
Flow direction
Area 1 = 400 ft²
Area 2 = 100 ft²
% flow exiting over
JK node link = 400 ft² / 500 ft² = 80 %
% of facets outflow exiting
over IJ node link = 100 ft² / 500 ft² = 20 %
Node J
Not To Scale

Figure 4—Schematic diagram of the routing of flows

CHECKR Algorithm

It is helpful in hydrologic modeling to know as much information about the watershed as possible. Examples of useful information are locations of stream lengths and ridge lines in the basin, since these influence the watershed's response to storm events. Also, knowing the locations of these entities would aid the user in validating the data set and in contributing to the GIS's completeness. The CHECKR algorithm used after PREPRO, is designed, using decision rules, to locate the ridge and stream lines in the watershed.

Determining the location of the stream and ridge lines in a watershed is facilitated by the nature of the TIN data structure and the information added to the database by the PREPRO algorithm. The rigid triangular structure of the TIN allows the decision rules to be simple and thus more easily coded. If the watershed were represented by a polygon structure, the location of ridge and stream lines would most likely come from another source, rather than an automated process as in the TINFLOW system.

The decision rule used to determine the location of a stream segment in the TIN involves examination of a facet's node link outflow statements. Each node link is examined, in turn, for a given facet. If the outflow statement for a given node link shows that it is not an output side for that facet, then that link is dropped from consideration as a stream segment since an inflow side cannot possibly be a stream segment. However, if that node link is an output side for that facet, then the record for the facet opposite that particular link is found through the TIN topology and brought into memory. Next, the outflow statement of the common node link is examined to determine if, for the new facet, it has also been classified as an outflow side. If the node link on the opposite facet is not an outflow side, then the flow is simply from one facet to another. But if this node link is also an outflow side , then a stream segment has been found, since two adjacent facets with a common outflow side define a stream segment. This node link's comment statement is then updated with this stream information. This process continues for each node link of each facet. After this operation is completed, sequential processing of each facet in the watershed resumes.

The decision rule used to determine the location of ridge lines in the TIN is similiar to the stream segment rule. Instead of searching for two adjacent outflow sides, however, two adjacent inflow sides are needed to classify the node link as a ridge segment. If two adjacent facets both classify the common link as an inflow side, then the flow is divided at this segment and is classified as a ridge line. This node link's comment statement is updated with this ridge information.

Because TINFLOW is a PC-based GIS, accessing random records in the relational database is the most time-consuming operation in the system. To decrease the time required to run the hydrologic model, the records in the database are put into a B-tree structure in the last step of the CHECKR algorithm. This step decreases the running time of the hydrologic model by approximately one half, even though a separate lookup file of TIN facet numbers must be accessed to tell the hydrologic model in which order the facets should be processed.

Hydrologic Processing

After the two preprocessors have prepared the database with the information and structure required by the hydrologic model, the model itself can be used. The hydrologic model component of the GIS simulates a watershed's response to storm events by modeling the two major components that contribute to storm runoff: overland flow and interflow.

Overland flow, as the name implies, is water that flows over the ground surface until it is intercepted by a stream segment. Interflow is water that infiltrates into the ground and then travels through the shallow soil layers to the stream. Ground water flow is not simulated in the model since it is generally considered to recharge the stream during periods of no precipitation, and does not usually contribute to storm runoff.

The TINFLOW system uses a finite difference solution of the St. Venant equations with a kinematic cascade approximation to simulate overland flow (Hong and Eli, 1985). For a finite difference solution to be used on a triangular element, the facet is converted during the simulation process to a rectangle with equal legnth and aspect ratios. The interflow process is modeled with the well-known Darcy's Law. The interflow model uses an expandable interflow zone, so that as the storm progresses, the wetting front of the interflow zone expands downward.

The TIN structure's capability of allowing attributes to be assigned on a facet-by-facet basis is vital to the hydrologic model. The attributes considered to have the greatest influence on simulating storm events are cover type and soil type. The cover type influences the amount of precipitation intercepted by vegetation before it reaches the ground. The soil type affects infiltration rates, roughness coefficients, and hydraulic conductivities.

The hydrologic model simulates, for an individual TIN facet, the hydrologic process for that land parcel. A hydrologic mass balance is calculated to determine if excess water is present and runoff can occur. Inputs to the mass balance are precipitation and water flowing onto the current TIN from uphill neighbor(s) through overland flow and interflow. To determine inputs, a record(s) of an uphill facet(s) is found by examining outflow percentages calculated during PREPRO. If the node link's outflow percentage is negative, flow exits over that node link. However, if the flow percentage is positive, flow enters current facet over that node link, and the uphill facet's outflow is accessed. If the current facet has two uphill neighbors contributing flow to the current facet, then the flow entering the current facet is a percentage of the outflows from the two uphill facets. Outputs include the overland flow and interflow exiting to the TIN facet downhill. The database is updated with these outputs so that they may be accessed by downgradient facets. This routing scheme requires that the processing begin with the most uphill facet in the watershed (i.e., the facet that has no uphill neighbors).

For a given time period, precipitation, if it occurs, is assumed to occur for the entire duration. After one period of precipitation has occurred, the water from that precipitation is routed through the basin by the sequential hydrologic processing of each facet. When a stream segment is encountered, as determined by CHECKR, the flow that would enter the stream is added cumulatively for the time period. The standard time period is one hour. It is assumed that, in a small watershed, any water that enters the stream during a given period will pass by the gage before the period ends.

Menu System

After the completion of a hydrologic simulation, a postprocessing menu of graphic outputs is presented to the user. Possible choices include a stream runoff hydrograph, a precipitation hyetograph or bar chart, a mass balance summary of overland flow, interflow and infiltration volumes, a cover type map, a soil type map, and a diagram of the watershed's stream network. Several of these graphics are illustrated at reduced scale in figures 5 through 7.

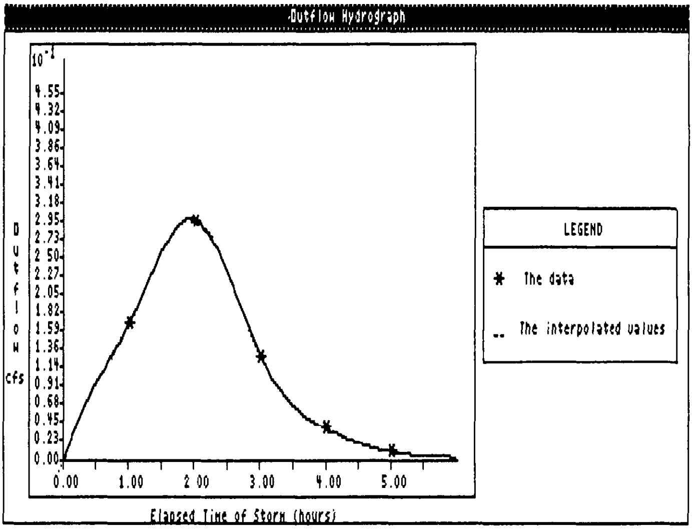


Figure 5—A runoff hydrograph

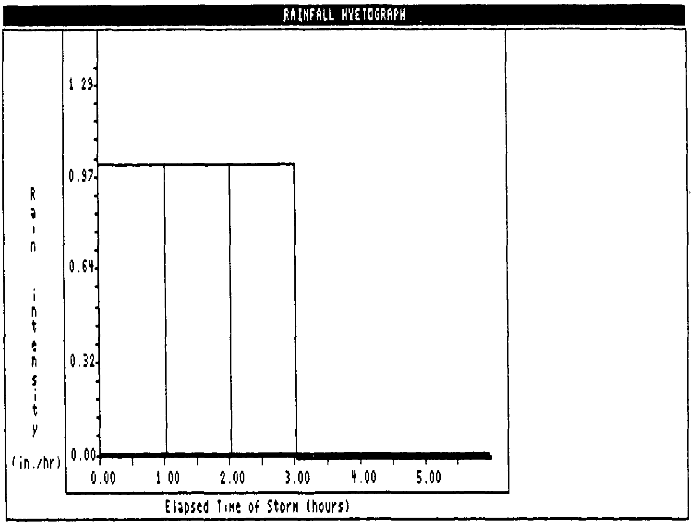


Figure 6—Precipitation hyetograph

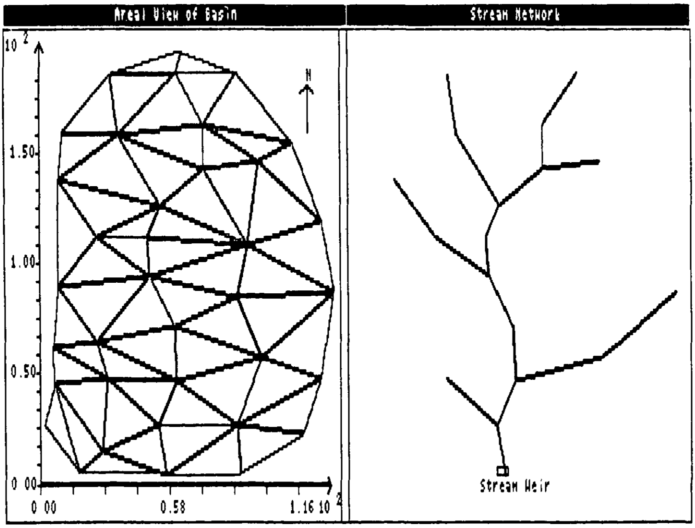


Figure 7—The watershed's stream network

CONCLUSIONS

The TINFLOW system is a functioning GIS that performs hydrologi c simulations on a personal computer. It has been tested on a synthetic watershed has yielded suitable results.

To improve the system, several avenues should be pursued. First, several of the assumptions made on the hydraulics of the hydrologic model should be examined and improved upon. Second, the quality of the outputs are limited primarily by the resolution of the PC graphics. The TURBO Pascal Graphics Toolbox used in the system supports a resolution of 640 x 200 pixels. Converting the system to a mainframe environment would allow faster processing times and provide the user with higher resolution screens on which to display graphic output. Third, the TINFLOW system's method of discretizing a watershed using the TIN should enable hydrologic simulations to be capable of predicting the change in a stream's runoff patterns caused by a change in a watershed. Demonstrating, on a real watershed, that the TINFLOW model could simulate a stream's storm hydrograph under natural conditions would be a large step forward for hydrologic modeling.

REFERENCES

Fowler, R.J. 1976, "Database Implementation for the TIN Data Structure", Technical Report 11, ONR Contract #n00014-75-c-088 6, Dept. of Geography, Simon Fraser Univ., B.C., Burnaby, Canada.

Hong, H.M. and Eli, R.N. 1985, "Accumulation of Streamflow in Complex Topography", Computer Applications in Water Resources, pp.196-205.

Linsely, Kohler and Paulhus 1975, Hydrology for Engineers, McGraw Hill, Inc.

Monmonier, M.S. 1982, Computer Assisted Cartography Principles and Prospects, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Peucker, T.K., Fowler, R.J., Little, J.J., and Mark, D.M., 1977, "Digital Representation of Three-Dimensional Surfaces by Triangulated Irregular Networks (TIN)", Technical Report 10, ONR Contract #N00014-75-c-0886, Dept. of Geography, Simon Fraser Univ., Burnaby, B.C., Canada.

Peucker, T.K. and Chrisman, N., 1975, "Cartographic Data Structures", American Cartographer, 2, no. 1, pp.55-69.

METHODS AND APPLICATIONS IN SURFACE DEPRESSION ANALYSIS

Susan K. Jenson*
TGS Technology, Inc.
Sioux Falls, South Dakota 57198

and

Charles M. Trautwein
U.S. Geological Survey
EROS Data Center
Sioux Falls, South Dakota 57198

ABSTRACT

Gridded surface data sets are often incorporated into digital data bases, but extracting information from the data sets requires specialized raster processing techniques different from those historically used on remotely sensed and thematic data. Frequently, the information desired of a gridded surface is directly related to the topologic peaks and pits of the surface. A method for isolating these peaks and pits has been developed, and two examples of its application are presented.

The perimeter of a pit feature is the highest-valued closed contour surrounding a minimum level. The method devised for finding all such contours is designed to operate on large raster surfaces. If the data are first inversely mapped, this algorithm will find surface peaks rather than pits.

In one example the depressions, or pits, expressed in Digital Elevation Model data, are hydrologically significant potholes. Measurement of their storage capacity is the objective. The potholes are found and labelled as polygons; their watershed boundaries are found and attributes are computed.

In the other example, geochemical surfaces, which were interpolated from chemical analyses of irregularly distributed stream sediment samples, were analyzed to determine the magnitude, morphology, and areal extent of peaks (geochemical anomalies).

* Work performed under U.S. Geological Survey contract no. 14-08-0001-20129.

Publication authorized by the Director, U.S. Geological Survey. Any use of trade names and trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the U.S. Geological Survey.

RATIONALE

Gridded surface data sets are critical components in many digital spatial databases. For example, Digital Elevation Model (DEM) data may be used to derive hydrologic information (Jenson, 1984), and gridded geochemical surfaces may be used to delineate areas of anomalous concentrations of a chemical element (Dwyer and others, 1984). However, while gridded surfaces are valuable datasets, they require analytical tools that recognize their special characteristics. While discontinuities may be present, gridded surfaces are discrete representations of primarily continuous data. The original control data that are used to generate a surface may be contour lines or control points, but the algorithms that compute the surfaces assume a continuous model. This underlying assumption distinguishes gridded surfaces from thematic spatial information such as digitized lithological units, and from remotely sensed data where manmade and natural discontinuities are frequent.

The information extracted from gridded surfaces is typically of a continuous nature, reflecting the data's origins and assumptions. For instance, slope and aspect information is commonly computed from DEM's, and directional derivatives are computed for geophysical surfaces. For visual information extraction, surface data are often represented using contour maps and mesh diagrams that aid interpretation of surface highs and lows.

The analytical tool presented here finds the topologic peak and pit polygonal features of a gridded surface. It is a specialized contouring process because the perimeters that define the peak and pit features are the lowest possible and highest possible closed contours, respectively, of the surface. Since these perimeters may occur at any data value in the data range, it is not possible to find these contours with standard algorithms without using an unreasonably small contour interval. This procedure was developed for hydrologic studies with DEM's, as in the DEM example presented later in the paper; however, it has utility for other types of surface data as well, as illustrated in the geochemical example.

An algorithm developed by Chan (1985) locates "lakes" in DEM data by locating a cell which may be within a depression and "growing" the lake with a stack-oriented algorithm. This approach requires an unacceptably large amount of computer memory to be allocated when depressions are large, as in some of the data presented here.

THE PROCEDURE

Since finding peak areas is the opposite process to finding pit areas, the peak- and pit-finding procedure was divided into two steps, both of which use the same computer programs. To find peak areas, the data are first inversely mapped. Either step or both steps may be selected for a given application. For purposes of describing the procedure, it will be assumed to be producing pit areas.

The objective of the procedure is to produce an output raster surface identical to the input raster surface, but with the cells contained in depressions raised to the lowest value on the rim of the depression. Therefore, each cell in the output image will have at least one monotonically decreasing path of cells leading to an edge of the data set. A path is composed of cells that are adjacent horizontally, vertically, or diagonally in the raster (eight-way connectedness) and that meet the steadily decreasing value criteria. If the input surface is subtracted from the output surface, each cell's resulting value is equal to its depth in a depression in the units of the input surface.

In order to accommodate large surfaces, the program was designed to operate in two modes. In the first mode, the surface is processed by finding and filling depressions wholely contained in 100-line by 100-sample blocks. In the second mode, the entire surface is processed iteratively in a circular buffer. Use of the first mode is memory intensive and the second is input and output intensive. A data set is first processed by the program in the first mode, thereby filling all depressions that do not intersect with cells with line or sample coordinates that are evenly divisible by 100. This intermediate data set is then processed by the program in the second mode to fill the remaining depressions. Processing in the second mode requires that only four lines of data be resident at any one time; therefore, large images can be processed. It is possible to further optimize the process for a given data set by varying the number of lines and samples processed in the first mode and to repeat the first mode with blocks staggered to overlie the join lines of the previous first mode pass. These modifications allow more of the depressions to be filled in the more efficient first mode. The procedure by which the first mode processes a block and the second mode processes the entire surface is the same for both modes and is as follows:

1. Mark all cells on the data set edges as having a path to the edge.

2. Mark all cells that are adjacent to marked cells and are equal or greater in value. Repeat this step until all possible cells have been marked.

3. Find and label all eight-way connected polygons of unmarked cells such that each polygon has maximum possible area. If no polygons are found, end the procedure.

4. For each polygon, record the value of the marked cell of lowest value that is adjacent to the polygon (threshold value).

5. For each polygon, for each cell in the polygon, if the cell has a value that is less than the polygon's threshold value, then raise the cell's value to the threshold value.

6. Repeat from step 2.

GEOCHEMICAL APPLICATION

An application to the detection and spatial characterization of geochemical anomalies that has been investigated demonstrates the utility of automated depression analysis techniques in the analysis of complex geochemical terrains. Geochemical anomalies, commonly defined by unusually high local concentrations of major, minor, and trace elements in rocks, sediments, soils, waters, and atmospheric and biologic materials, are important features in studies related to mineral and energy resource exploration and environmental monitoring. These anomalies are usually detected by establishing a threshold concentration that marks the lower bound of the anomalous concentration range for each element in each type of material. The threshold value is used to sort the geochemical data into background and anomalous sample populations, which then may be plotted on a map for comparison with other data. For certain types of materials and terrains in which background concentrations for selected elements are relatively uniform, this approach is satisfactory. However, in geochemical terrains where background values are variable across the region studied, this approach is commonly modified by removing regional trends prior to the selection of an appropriate threshold value.

Trend-surface analysis is frequently used to mathematically model regional variations in geochemical data sets. In this technique, first-, second-, and higher-order equations are used to describe regional trends in terms of the data set's best least-squares fit to planar, parabolic, and higher order nonplanar surfaces. The resultant regional model is subtracted from the original data leaving residual concentrations that represent local variations, above and below, the regional trend. Positive variations are then statistically evaluated to establish a threshold. This procedure works well in areas where regional controls, and their consequent effects, are known; however, in most areas trend-surface analysis only provides an approximation of an unknown function with an arbitrary, best-fit function.

Because many types of geochemical data are cartographically represented as contour maps (with contour intervals equated to chemical concentration ranges) and geochemical anomalies are topologically analogous to localized peaks on a topographic map, automated depression analysis techniques were applied to a rasterized geochemical data set in an effort to more objectively define anomalies based on their morphology.

A geochemical data set was studied that consisted of 2,639 analyses of copper concentration in the heavy mineral fraction of stream sediment samples distributed throughout the Butte 1° x 2° Quadrangle, Montana. The analyses, which were referenced by latitudes and longitudes of the sample collection sites, were rasterized using a minimum curvature interpolation and gridding algorithm (Briggs, 1977). The resultant grid consisted of a 559- by 775-cell array of 200-meter by 200-meter (ground-equivalent size) grid cells cast in a Transverse Mercator map projection. Interpolated copper concentration values in the array were in the range from 0 to 65,684 ppm (parts per million) copper with an arithmetic mean of 161.17 ppm. Figures 1 and 2 show the distribution of original sample sites within the quadrangle and a grey-level representation of the interpolated concentration surface.

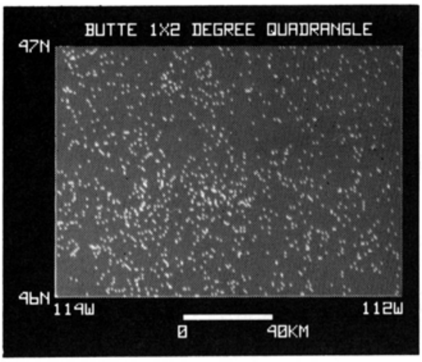


Figure 1.--Distribution of geochemical sample sites.

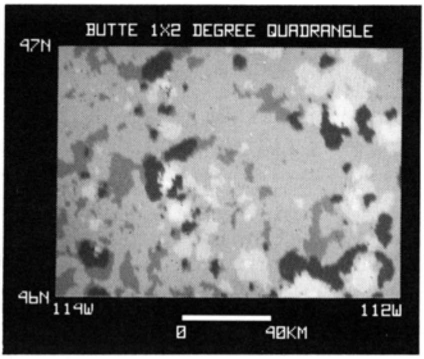


Figure 2.--Gray-level map of copper surface; brighter tones represent higher concentration intervals.

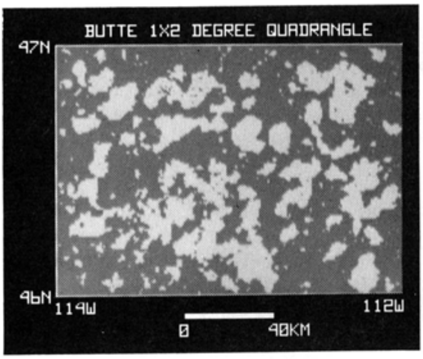


Figure 3.--Topologically defined copper anomalies.

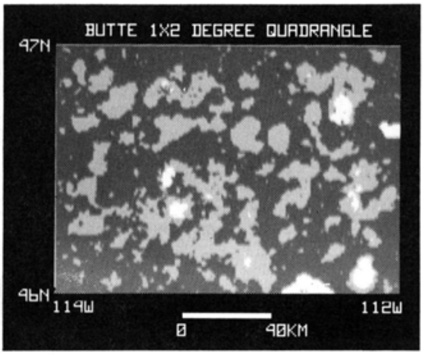


Figure 4.--Comparison of topologically defined copper anomalies (gray) and copper anomalies defined by a 1,000 ppm threshold (white).

A mapping function was used to topographically invert the interpolated range of values. The product of this operation was then subjected to the depression analysis algorithm described earlier. Anomalies, in their inverted form, are morphologically described through this algorithm as closed depressions. The depressions found in the inverted data for the Butte quadrangle are shown in gray in figure 3. In figure 4, these same depressions are shown in gray again, and superimposed in white are the areas that are above a 1,000 ppm threshhold. The white areas are the only areas that are identified by a traditional single-threshhold approach. This comparison demonstrates the utility of the morphologic approach in areas such as this where regional variations are extreme and a single threshold value is insufficient for detecting anomalies in different parts of the geochemical terrain. While the morphologic approach identifies many more potentially anomalous areas, more analysis is required to relate the ppm values in the area to the area's background material types. An additional advantage of this approach is that it does not require generation of a separate, often arbitrary, model of the regional trend as in cases where trend-surface analysis is performed.

A visual comparison of peaks and the control points that are within them or nearby them is beneficial in that each peak's reliability can be evaluated. If many control points appear to be defining a peak, the analyst may feel more confident in categorizing that peak as anomalously high. However, if the control points are few or badly distributed, the peak may be categorized as an overshoot in the surface-generation process.

DEM HYDROLOGY APPLICATION

The National Mapping Division and Water Resources Division of the U.S. Geological Survey cooperated with the Bureau of Reclamation in 1985 and 1986 to objectively quantify and to incorporate the contributing and noncontributing factors of pothole terrain in a probable maximum flood estimate for the James River Basin above the dam at Jamestown, North Dakota. The hydrology of the area has been difficult to study due to flat slopes, the complex nested drainage of the potholes, and a poorly defined drainage network.

DEM's were used to derive hydrologic characteristics that were incorporated in rainfall runoff models. DEM's were made for five test sites in the Basin. Each test site covered approximately 10 square miles with a 50- by 50-foot grid-cell size. The largest DEM was 505 lines by 394 samples.

For each test site, the surface depression procedure was the beginning step for the DEM analysis. Once the surface depressions were identified, they were given unique identifying labels and their volumes were calculated. A subset of depressions were selected for the modeling process based on a minimum volume criteria. Some depressions that did not meet the volume criteria were still modelled because they were spatially necessary to complete drainage linkages.

A second processing step then found the watershed boundaries for these selected depressions. A previous watershed program (Jenson, 1984) had to deal with real and artificial depressions in the paths of drainages by running iteratively and using thresholds to "jump" out of holes. By taking advantage of the depression map of the surface, however, the watershed program could be modified to run in two passes. The surface processed by the watershed program was the surface with all depressions filled except those that were selected for the modeling process. A shaded-relief representation for the DEM of one of the test sites is shown in figure 5. The corresponding selected potholes and watershed boundaries are shown in figure 6.

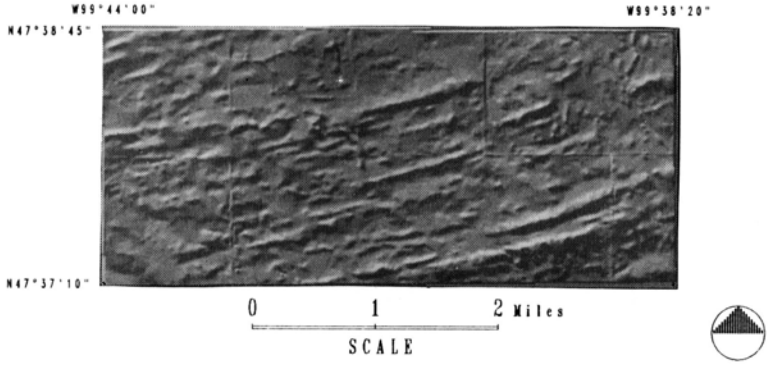


Figure 5.--Shaded-relief representation of digital elevation model data for one of the James River Basin test sites.

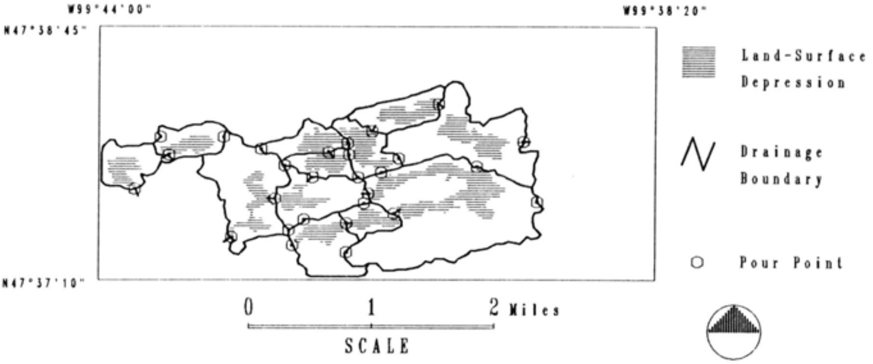


Figure 6.--Selected potholes, watersheds, and pour points for the test site in figure 5.

CONCLUSIONS

This depression-finding procedure has been shown to be practical and useful in the analysis of geochemical and DEM surface data sets. For inversely-mapped geochemical surfaces, depression analysis indicates areas of anomalously high chemical concentrations and bypasses the need for trend surface analysis. The hydrologic analysis of DEM surfaces benefits from depression identification because depressions may be hydrologically significant themselves, such as potholes, and the removal of unwanted depressions simplifies the automated finding of watershed boundaries.

REFERENCES

Briggs, I. C., 1977, Machine contouring using minimum curvature, Geophysics, vol. 39, no. 1, p. 39-48.

Chan, K. K. L., 1985, Locating "lakes" on digital terrain model; Proceedings, 1985 ACSM-ASPRS Fall Convention, p. 68-77.

Dwyer, J. L., Fosnight, E. A., and Hastings, D. A., 1984, Development and implementation of a digital geologic database for petroleum exploration in the Vernal Quadrangle, Utah-Colorado, U.S.A.; Proceedings, International Symposium on Remote Sensing of Environment, p. 461-475.

Jenson, S. K., 1984, Automated derivation of hydrologic basin characteristics from digital elevation model data; Proceedings, Auto-Carto 7, p. 301-310.

MULTIPLE SOURCES OF SPATIAL VARIATION AND HOW TO DEAL WITH THEM.

P.A. Burrough
Instituut voor Ruimtelijke Onderzoek
University of Utrecht,
Postbox 80.115, 3508 TC Utrecht,
The Netherlands

ABSTRACT

Conventional methods of thematic mapping often assume implicitly that only one major pattern can be recognized at any given scale of mapping. Conventional thematic map representations model spatial units by 'homogeneous' units or polygons representing the various components of the pattern being mapped. Interpolation methods allow gradual variation within spatial units to be mapped but they commonly also ignore the problems that arise from multiscale sources of variation. Observed natural variation may be caused by a number of separate spatial processes operating with various weights (intensities) over a range of scales. This paper reviews some ways in which theoretical multiscale models, complex semivariograms, robust methods and sampling strategies can be applied to the problem of multiple sources of spatial variation.

INTRODUCTION

The search for quick, cheap, simple, reliable and universal ways with which to capture and describe the spatial variation of attributes of the natural environment is a current major research activity. There are many ways to describe and map the spatial variation of soil, vegetation, landform, groundwater or pollution. Some researchers follow the well-worn paths of tried and tested methods while others strike out through thorny, mathematically difficult terrain. In spite of many, local near successes, and many global failures, the search for useful, reliable methods of spatial analysis continues unabated across all disciplines whose object it is to study the spatial variation of attributes of the earth's surface. Considering the costs involved in collecting and analysing spatial data, and the implications for landuse planning decisions of incorporating poor or incorrect data in geographical information systems, it is crucially important for data users to know how spatial data have been modelled, and what the limitations of these models are. One limitation that is frequently overlooked when choosing an interpolation method is the presence of important variation at several scales which may confound or reduce the success of the chosen spatial modelling technique.

Methods for spatial analysis

The two basic approaches to mapping the spatial distribution of any given attribute, or regionalized variable (Matheron 1971) are summarized in Table 1. In the first approach one has total coverage of an area, usually with remotely sensed imagery (aerial photos or digital scanned images) of an attribute or attributes that are thought to be correlated with the required environmental property. In the second approach one samples the property of interest directly at certain locations from which a model of the spatial variation is created by interpolation.

Table 1. Basic approaches to mapping

Whole area approach

- Many observations of cheap, possibly relevant data.
- Divide area into regular units (pixels) or into 'natural' units
- Devise and use hierarchical classification schemes
- Discover relations between attribute values of pixels or class means of 'natural' units and attribute of interest.

Point sampling approach.

- Choose sampling strategy (regular grid, stratified random, etc.)
- Choose and apply interpolation method (global, local, etc.).
- Map isolines

MATHEMATICAL MODELS OF SPATIAL VARIATION

The classificatory, choropleth map model approach relies on the model

$$Z(x) = \mu + \alpha_j + \epsilon \qquad (1)$$

Where $Z(x)$ is the value of attribute Z at point x, μ is the general mean of the area in question, α_j is the deviation between the mean of class j and μ, and ϵ is the residual variation, usually assumed in the first instance to be a normally distributed Gaussian noise function having zero mean and variance σ^2. The weakness of this model is revealed every time an area is remapped at a larger scale, thereby 'discovering' spatial structure in what was previously regarded as spatially unstructured and uncorrelated 'noise'. As this process of remapping at larger and larger scales can continue endlessly, the success of this mapping approach depends greatly on the balance between the different kinds and scales of spatial variation present. The universal nature of this problem is revealed by studies that show that irrespective of map scale, the distribution of boundaries on thematic choropleth maps over a wide range of scales can be modelled satisfactorally by a Poisson distribution

$$P(x) = 1 - \exp(-\lambda x) \qquad (2)$$

or related functions such as the Gamma distribution or the Weibull function (Burgess and Webster 1984, Burrough 1986)

Short-range variation in digital imagery.

The presence of short-range variation in digital imagery is usually considered a nuisance that needs to be removed. If the source of the noise is known, many techniques exist for its removal (e.g. destriping LANDSAT images). If the source is unknown, but local, simple digital filter techniques exist for mechanistic removal of the unwanted noise (c.f. Rosenfeld and Kak 1976). Statistical methods of image analysis, recently reviewed by Ripley (1986) also assume that at the chosen observation scale a clear signal is waiting to be cleaned up (see also Besag 1987).

Methods of interpolation.

In many situations such as in studies of soil fertility or pollution, it is impossible or impractical to obtain a complete overview using surrogate attributes and so the phenomenon of interest must be mapped using samples collected at point locations. The overall distribution of the variation of the phenomenon is then determined by interpolation. Methods of spatial interpolation (c.f. Agterberg 1982, Burrough 1986, Davis 1986, Lam 1983, Ripley 1981) adopt either a global or a local approach. Global methods, such as trend surface analysis, parallel choropleth map models in the sense that they attempt to 'explain' large amounts of spatial variation in terms of single structural units (complex polynomials). Just as with the choropleth map models, the 'noise' usually contains short-range spatially correlated variation. Local methods avoid these problems, but introduce others, such as how best to choose the local weighting function and how to select the most appropriate method of interpolation (e.g. smooth B-splines or moving weighted averages).

Optimal methods of interpolation (kriging).

The set of interpolation techniques collectively known as kriging recognise that spatial variation may be the result of structural, locally random but spatially correlated, and uncorrelated components. Information about these various components is used to compute the weights for local interpolation in such a way as to minimize the variance of the interpolation estimate. The basic model is:

$$Z(x) = m(x) + \epsilon'(x) + \epsilon'' \qquad (3)$$

in which the value of attribute Z at point x is modelled by m(x), a deterministic function describing the 'structural' component of variation, $\epsilon'(x)$ is a function describing the local, spatially correlated variation of Z, and ϵ'' is a random noise term. The essential steps in kriging (Journel and Huijbregts 1978, Webster 1985) are:

1. Sampling to determine the sample semivariogram
2. Fitting an appropriate model to the sample semivariogram
3. Using the semivariogram model to supply appropriate values of the weights with which to obtain estimates of the value of Z at unvisited points x0.

MULTIPLE SCALES OF VARIATION AND KRIGING

Kriging is a practical and a conceptual advance on previous methods of spatial interpolation because it allows 'non-structural' variation to be considered as being comprised of spatially correlated variation and random variation. The critical aspects of kriging, however, are the fundamental assumptions of the method and the choice and fitting of semivariogram models. In both instances, the type and nature of multiscale variation can be critically important.

The fundamental assumptions of kriging are contained in the intrinsic hypothesis of regionalized variable theory which regards spatial variation as the outcome of a random process with certain stationarity conditions. These are:

1. That the expected difference in the value of Z at any two places separated distance h is zero:

$$E[Z(x) - Z(x+h)] = 0 \qquad (4)$$

2. the variance of the differences depends on h and not on x, and is given by:

$$\begin{aligned} \mathrm{var}[Z(x) - Z(x+h)] &= E[\{Z(x) - Z(x+h)\}^2] \\ &= 2\,\gamma(h) \qquad (5) \end{aligned}$$

Clearly, these assumptions require that the spatial process in question operates over the whole of the area to which consideration is being given.

The semivariogram and semivariogram models.

The semivariogram displays the variation of semivariance with sample spacing, h. It is obtained by sampling and through the intrinsic hypothesis it is estimated by

$$\hat{\gamma}(h) = \frac{1}{2n(h)} \cdot \sum_{i=1}^{n(h)} \{z(x_i) - z(x_i + h)\}^2 \qquad (6)$$

where n(h) ıs the number of pairs of observations with separation h.

Usually, the weights for interpolation are obtained by fitting a suitable model to the experimentally estimated semivariances. Two major classes of semivariogram model have been recognised: a) the transitive models; b) unbounded models.

Because of the variance of the estimate Z at any point can not be less than zero, the sample semivariogram cannot be modelled by any function that appears to fit the distribution of points. The following ***authorized models*** are recommended for use (McBratney and Webster 1986):

a) transitive models - i.e. models in which the semivariance appears to reach a constant level (the sill) at a certain sample spacing or range:

linear model with sill	(1D only)
circular model	(1D, 2D)
spherical model	(2D, 3D)
gaussian model	(1D, 2D)
exponential model	(1D, 2D, 3D)

b) unbounded models - i.e. models in which the semivariance continues to increase with sample spacing:

linear model	(1D, 2D, 3D)
logarithmic model	(1D, 2D, 3D)
brownian fractal model	(1D, 2D, 3D)

Multiscale variation.

All transitive models, with the exception of the exponential model, imply that the observed variation has been generated by a spatial process that operates at a definite scale, for example within overlapping blocks that have a definite size or scale. Under these circumstances the spatial model given by equation (3) describes the situation adequately. With the exponential model, and the unbounded models, however, it is implicit that variations can occur over a range of scales. The exponential model suggests that the overlapping blocks vary randomly in size; the unbounded models, particularly the fractal and the logarithmic model, suggest that spatial variation occurs at many scales. A semivariogram that approaches the origin parabolically may signify changing drift (i.e. change in the value of E[Z(x)] with x caused by local or regional trends - i.e. variation at another scale). Changing drift can be handled either by using a full structural analysis and universal kriging as described by Olea (1975), or by using intrinsic random functions of a higher order that the semivariogram to describe the spatial variation (Matheron 1973).

Choosing the correct semivariogram model is critical for kriging, yet little attention seems to have been paid to the physical grounds for choosing any particular model. There are several aspects of the problem. The first is the nature of the variation being studied - is it the result of a single, dominant process or the sum result of several superimposed processes? What kind of spatial distribution results from a given physical process? The second is the problem of sampling variation on the estimated semivariogram - how much can the form of a semivariogram vary according to the sample of points used? The third is the problem of the choice and fitting of models, and whether that choice should be guided primarily by least-squared fit criteria or by using other criteria.

A simple multiscale model. Instead of considering that observed spatial variation is the result of structural, local randomly correlated and random components as expressed by equation (3), let us now assume that randomly correlated variation can exist at all scales. Mandelbrot's Brownian fractal model (Mandelbrot 1982) is the ideal embodiment of a model in which spatial variation occurs at all scales. The simple Brownian model has several draw- backs in practice, however; it assumes that variation occurs at all scales in a self-similar way, and that the roughness of the variation (the value of the D parameter) is the same at all scales. Consideration of real data suggests otherwise (Armstrong 1986, Burrough 1984). Real spatial processes (omitting special cases such as cloud formation) seem to lead to spatial patterns in which the fractal D value varies with location and with scale (Mark and Aronson 1984).

With this in mind, I developed a one-dimensional nested model of spatial variation that is an extension of equation (3), but within which the scales and the weights of the various components can be set independently (Burrough 1983). The value of Z at point x is now given by

$$Z(x) = \sum_{i=1}^{n} \{ \epsilon'_i(x) \} + \epsilon'' \qquad (7)$$

where the $\epsilon'(x)$ are a set of nested, spatially correlated random functions associated with scale i. As before, the ϵ'' term represents spatially uncorrelated random variation to take account of measurement errors and other essentially random, non-spatial sources of variation.

The model has since been programmed for interactive use as a personal computer 'game' and it allows the user to create one-dimensional displays of multiscale data by setting the ranges and weights of several nested random functions. The semivariogram is displayed together with the function (Figure 1). The computer game has proved invaluable for teaching students and others not familiar with spatial statistics how complex spatial variation can arise from nested random processes, and also for demonstrating the problems associated with under-sampling. The game allows transects from 20 to 600 points to be generated. Generating the same model several times for different transect lengths allows the user to see how an estimate of a semivariogram relies on sufficient samples.

If one can generate a transect from single random processes, it should be possible, in principle, to go the other way and to estimate the scales and weights of the contributing processes from the sample semivariogram. Simple geological transects gave good results (Burrough 1983), with the valuable by-product that the confidence limits and effective degrees of freedom of the fitted model could be calculated (Taylor and Burrough 1986; see also McBratney and Webster 1986). Alas, preliminary results of work with two-dimensional simulations suggest that decomposing multi-scale two-dimensional patterns is not so straightforward.

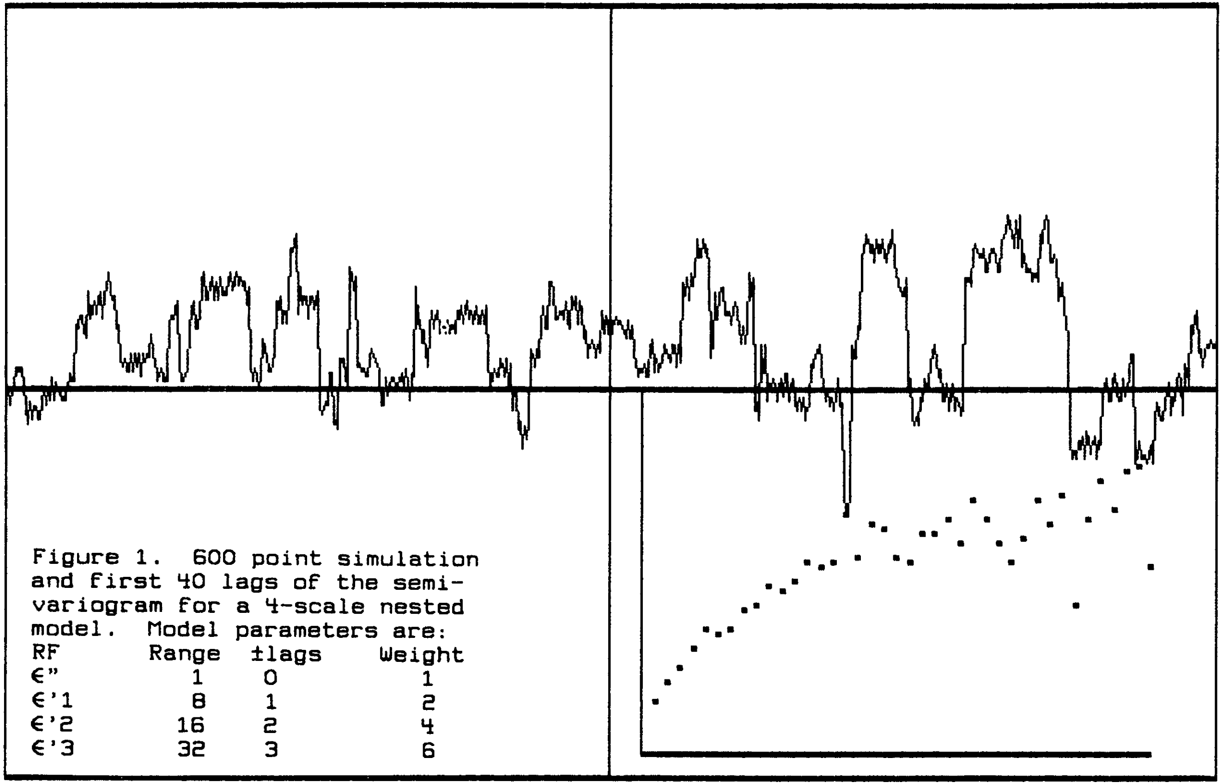


Figure 1. 600 point simulation and first 40 lags of the semi-variogram for a 4-scale nested model. Model parameters are:

RF	Range	±lags	Weight
€"	1	0	1
€'1	8	1	2
€'2	16	2	4
€'3	32	3	6

Complex multiscale models. The one-dimensional nested model is only authorized for work in one dimension, so the approach must be modified when working in two or more dimensions. An alternative to fitting a single, complex model is to choose several standard authorized models and to combine them to give an overall, complex model. The question then is on what grounds the separate models should be chosen. McBratney and Webster (1986) demonstrate the use of double models for semi-periodic soil variation in Australian gilgai, and for heavy metal concentration in soil in Scotland. In both cases they made use of their knowledge about the physical soil processes to guide their choice of the components of the model. As with all models, the investigator needs to strike a balance between goodness of fit to the data and parsimony. McBratney and Webster (1986) suggest that the choice between a single scale model and a multiscale model (or between two multiscale models) can be estimated by using Akaike's (1973) information criterion which is estimated by

$$\hat{A} = n \ln(R) + 2p \qquad (8)$$

where n is the number of observations, p is the number of estimated parameters and R is the residual sum of squares of the fitted model. The model with lowest $\hat{A}$ is the best. Here I should like to remark that it is possible that the best fitting model may not always make physical sense. For example, if a best-fitting semivariogram model returns an estimate of the nugget variance €" that is considerably less than that known to be possible with the given laboratory technique, the results should be treated with caution.

Robust methods of estimating the semivariogram

When an essentially point process is superimposed upon a continuous process, estimates of the semivariogram obtained by equation (6) may be heavy tailed because the intrinsic hypothesis is locally invalid. McBratney and Webster (op cit.) cite this problem when mapping soil potassium over a cow pasture contaminated with faeces; we have noted similar problems in cracking clay soils in the Sudan and in soil pollution (Rang et al 1987). Cressie and Hawkins (1980) proposed robust methods to deal with the problem of heavy-tailed distributions; McBratney and Webster (op cit.) suggest that the robust methods are of most value when an underlying spatial process needs to be separated from the effects of a contaminating point process.

DISCUSSION AND CONCLUSIONS

Most natural patterns of variation contain contributions from processes operating at various scales. When a particular scale of variation is dominant and obvious, standard mapping techniques will often suffice. When several scales are important, it may be necessary to identify them before proceeding further, using all available knowledge about the processes in question in order to make sensible decisions.

Separation into 'natural' physiographic units may be a wise first move that can ensure that the basic assumptions of a mapping technique hold throughout a single area (e.g. see Burrough 1986). Knowledge of spatial processes and the patterns they are likely to create may also assist when choosing both simple and complex models. The definite choice of complex models and the estimation of relative weights and scales of variation is made difficult by uncertainties in the estimation of semivariograms.

One way to avoid capturing too many levels of spatial variation is by tailoring sample spacing before mapping. There is now considerable evidence (e.g. Oliver and Webster 1986, Webster 1985) that nested methods of sampling can provide useful estimates of the scales of spatial variation present in an area before mapping or sampling for the semivariogram commences.

REFERENCES

Agterberg, F.D. 1982. Recent developments in GeoMathematics. Geo-Processing 2, 1-32.

Akaike, H. 1973, Information theory and an extension of maximum likelihood principle. In: Second International Symposium on Information Theory. (Eds. B.N. Petrov and F. Coaki) pp. 267-281, Akademia Kiado, Budapest.

Armstrong, A.C. 1986, On the fractal dimensions of some transient soil properties. J. Soil Sci. 37, 641-651.

Besag, J. 1987. On the statistical analysis of dirty pictures. J. Royal Statistical Soc. Section B. (in press).

Burgess, T.M. and Webster, R. 1984, Optimal sampling strategies for mapping soil types. I. Distribution of boundary spacings. J. Soil Sci. 32, 643-659.

Burrough, P.A.. 1983, Multi-scale sources of spatial variation in soil. II. A non-Brownian Fractal model and its application to soil survey. J. Soil Sci. 34,599-620.

Burrough, P.A. 1984, The application of Fractal ideas to geophysical phenomena. Bull. Inst. Mathematics and its Applications. 20, 36-42.

Cressie, N. and Hawkins, D.M. 1980. Robust estimation of the variogram. Math. Geology 12, 115-125.

Davis, J.C. 1986. Statistics and Data Analysis in Geology. Wiley (2nd. Edn).

Journel, A.J. and Huijbregts, Ch. J. 1978. Mining Geostatistics. Academic Press.

Lam, N. S., 1983. Spatial interpolation methods: a review. The American Cartographer 10, 129-149.

Mandelbrot, B.B., 1982. The Fractal Geometry of Nature. Freeman, New York.

Mark, D.M. and Aronson, P.B., 1984. Scale-dependent fractal dimensions of topographic surfaces: an empirical investigation with application in geomorphology and computer mapping. Mathematical Geology, 16, 671-83.

Matheron, G., 1971, The theory of regionalized variables and its applications. Cahiers du Centre de Morphologie Mathêmatique de Fontainebleu, No. 5, Paris.

Matheron, G. 1973. The intrinsic random functions and their applications. Adv. Appl. Prob. 5, 439-468.

McBratney, A.B. and Webster, R. 1986, Choosing functions for semivariograms of soil properties and fitting them to sample estimates. J. Soil Sci. 37, 617-639.

Olea, R.A. 1975, Optimum mapping techniques using regionalized variable theory. Series on Spatial Analysis No. 2, Kansas Geological Survey, Lawrence.

Oliver, M.A. and Webster, R., 1986. Combining nested and linear sampling for determining the scale and form of spatial variation of regionalized variables. Geographical Analysis 18, 227-242.

Rang, M.C., Ockx, J, Hazelhoff, L. and Burrough, P.A. 1987, Geostatistical methods for mapping environmental pollution. Paper presented Int. Symposium on Soil and Groundwater Pollution, Noorwijkerhout, The Netherlands, 30 March-2 April 1987.

Ripley, B. 1981. Spatial Statistics, Wiley, New York.

Ripley, B. 1986. Statistics, Images and Pattern Recognition. Canadian J. Statistics 14(2), 83-111.

Rosenfeld, A. and Kak, A. 1976. Digital Picture Processing. Academic Press, New York.

Taylor, C.C. and Burrough, P.A., 1986. Multiscale sources of spatial variation in soil III. Improved methods for fitting the nested model to one-dimensional semivariograms. Math. Geology 18, 811-821.

Webster, R. 1985. Quantitative Spatial Analysis of Soil in the field. Advances in Soil Science Volume 3, Springer-Verlag New York.

A CARTOGRAPHER'S APPROACH TO QUANTITATIVE MAPPING OF SPATIAL VARIABILITY FROM GROUND SAMPLING TO REMOTE SENSING OF SOILS

Ferenc CSILLAG

Research Institute for Soil Science and Agricultural
Chemistry of the Hungarian Academy of Sciences
MTA TAKI, Budapest, Herman Ottó u. 15. H-1022

ABSTRACT

Data collection and the handling of spatial information inherent in the data for natural resource mapping is cumbersome and particularly problematic in large area surveys for which remote sensing provides an excellent tool in resource assessment and process monitoring. In this case ground sampling is used not only to produce maps but also to calibrate remotely sensed data, therefore the statistical and physical relationships between them should be treated quantitatively, as well as the features of the resulting thematic maps. This approach can be extensively used in both ways: in sampling design and in accuracy testing.

From a theoretical point of view the sampling design can be well optimized in terms of sample size, acceptable error and confidence limits to describe the spatial variability of Earth surface features, when measurment errors and characteristics of spatial patterns are taken into consideration. This strategy should be also applied for remotely sensed data.

Once spatial data is constructed either by interpolation or remote sensing, thematic maps are generally derived. A geographic expert system should additionally exploit in this processing the understanding of errors related to the geometric and thematic determination of contours.

INTRODUCTION TO THE IDEAS

The last decade has produced and spread a number of new data sources, consequently processing technology in spatial data handling. Geographers, cartographers, remote sensing scientists and others have been working on the exploitation of these opportunities in quantitative resource mapping. The impact of this development on cartography was well demonstrated, among others, at the previous AutoCarto Conference in London (e.g. Morrison 1986).

A number of authors have proved the potential of sampling theory applied for mapping, in particular in experiment design (McBratney and Webster 1981), calibration of remotely sensed data (Curran and Williamson 1986a) and thematic map accuracy testing (Rosenfield et al. 1982), to provide tools for determining strategies of better sampling to achieve better accuracy for, in general, better (i.e. quantitative) description and understanding of spatial phenomena.

One of the most popular and important keywords for those involved in this type of research is spatial variability. The basic idea of this paper is to raise and outline the issue of incorporating our knowledge of spatial variability in geographic informations systems (GIS), thus developing it to a geographic expert system (GES). The approach is thought to be appropriate for a wide variety of applications in both sampling design and in accuracy testing.

Part one is a compact summary of sampling theory with regard to ideal vs. field sampling and remote sensing. It is followed by a section on how maps can be constructed from raw data with regard to spatial variability of the represented variables. Then some aspects of contour definition for thematic maps are outlined and finally an ongoing experimental soil mapping experiment is presented with preliminary results.

SAMPLING REVISITED

Sampling theorem for stationary processes

Many properties of several Earth surface features can be treated as continuous signals, however, scientists need to describe their patterns using generally sparse point observations. Field sampling in this respect can be represented as seen in Fig.1(a-c). Having the continuous signal g(x), where x denotes spatial coordinate(s), point sampling can be approximated with a Dirac-δ series:

$$g(x)\Sigma_k\delta(x-kd)=\Sigma_k g(x)\delta(x-kd) \qquad /1/$$

where d denotes the sampling distance. From a number of illustrative experience one can have the impression that the shorter the sampling interval, the better the reconstruction of the signal, although the right hand side of Eq./1/ is equal to zero at every $x \neq kd$.

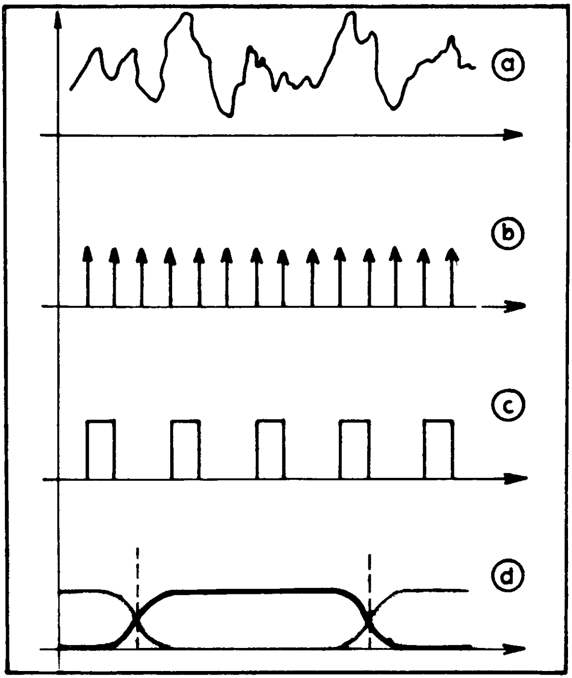


Figure 1.

Schematic representation of a property as a continuous signal (a) and its ideal (b), real in-situ (c) and remote (d) sampling.

The exact relationship between the sampled and original signal is defined by the sampling theorem (see e.g. Meskó 1984). Recalling the Fourier-transform of the Dirac-δ series, the Fourier-transform of Eq./1/ can be written as

$$G(f)*(1/d)\Sigma_k\delta[f-(k/d)]=(1/d)\Sigma_k G[f-(k/d)] \qquad /2/$$

where * denotes convolution, f stands for (spatial) frequency and the expression gives the sampled spectrum $G_S(f)$. The summed terms of the right hand side may be overlapping, however,

$$\text{if } |f|>f_{UP}\leq f_{NYQUIST}=1/2d$$
$$\text{then } G(f)=0 \text{ for } |f|>f_{UP} \qquad /3/$$

meaning that the original signal can be reconstructed without any loss of information if the period of the highest spatial frequency is, at least, twice sampled.

Once having the Fourier-transform of g(x), the autocovariance function can be derived as follows:

$$COV(h)=F^{-1}\{|G(f)|^2\} \qquad /4/$$

from which spatial patterns (e.g. dominant frequency) can be computed.

Quantitative description of variability of not strictly stationary processes

Eq./4/ implies that our process for which the g(x) signal is recorded should be stationary in both mean and variance. As it happens many properties of the land appear not to be stationary in this sense (Oliver and Webster 1986). This led Matheron (1965) to consider the somewhat weaker assumptions of stationarity:

$$E\{g(x)-g(x+h)\}=0 \qquad /5/$$

and

$$VAR\{g(x)-g(x+h)\}=E\{[g(x)-g(x+h)]^2\}=2\mu(h) \qquad /6/$$

where E denotes expectation and the function $\mu(h)$ is called the semi-variogram.

(If the process is second order stationary then than the semi-variogram is related to the autocovariance function by: $\mu(h)=COV(0)-COV(h)$, and either $\mu(h)$ or COV(h) can be used to describe the spatial process. If, however, only the so called intrinsic hypothesis holds (c.f. Eqs./5/ and /6/), than the covariance is undefined and g(x) is called a regionalized variable. The semi-variance can be estimated without bias according to the definition implicit in Eq./6/, or to the formulas for two or more dimensions and irregular sampling (Webster 1985).)

FROM DATA TO MAPS

Optimal interpolation and estimation for spatial units

The method of estimation embodied in regionalized variable theory is known in Earth sciences as kriging (Webster 1985). It is essentially a means of weighted local averiging:

$$g'(y)=\Sigma_k L_k g(x_k) \qquad /7/$$

in which the weights (L_k) are chosen so as to give unbiased estimates at y ($g'(y)$).

It is optimal in the sense, that at the same time it minimizes the estimation variance:

$$\sigma_E^2(y)=E\{[g(y)-g'(y)]^2\}=$$

$$=2\Sigma_k L_k \mu^*(x_k y)-\Sigma_k \Sigma_m L_k L_m \mu(x_k x_m)-\mu^*(y) \qquad /8/$$

where $\sigma_E(y)$ is the estimation variance at y (that can be either a point or a block), E denotes expectation, $\mu(x_k y)$ stands for the semi-variance of the property between x_k and y, taking into account of both the distance and angle, while $\mu^*(x_k y)$ and $\mu^*(y)$ denotes the average semi-variance between x_k and all points within the block, and within the block, respectively. (For formulas to obtain the estimation variance see Webster 1985.)

Application to large area surveys, remote sensing and GIS

Large area surveys of natural resources, in general, require an enormous amount of samples when applying systhematic sampling. Remote sensing techniques, however, confine ground sampling to training areas only. Regionalized variable theory (or otherwise geostatistics) can reduce the necessary sample size to even less, and provide information on the accuracy of a constructable map as a function of location while raising the efficiency of data processing in the following ways:

(1) To obtain the semi-variogram from ground observations to describe the scale and patterns of spatial variables over several orders of magnitude nested sampling is a very economic way (Oliver and Webster 1986);

(2) Once the semi-variogram is obtained the spatial resolution with acceptable estimation error, or, conversly, the error for a given spatial resolution for a given property can be determined;

(3) This can be then compared to the optimum spatial resolution derived from remote sensing pilot studies of "homogenity" (Curran and Williamson 1986b);

(4) Finally structural information of the remotely sensed data themselves can be used in digital image processing (Carr and Myers 1984).

From the above listed aspects (2) is in the focus of our interest, because it is not only applicable for calibration purposes with ground and remotely sensed data, but can be introduced as an expert system function in a GIS. Thematic accuracy have not only been neglected untill now in standard GIS products, but "intelligent map edition" may serve as a permanent temptation for constructing meaningless but nice graphic products from a very limited number of samples. Even with sufficient data, accuracy as a function of location can be used as an overlay or auxilliary information in a GIS for data representation and/or for further processing (like scale chenges etc.), providing a quantitative cartographic tool for deeper understanding of spatial features.

MAPS AND CONTOURS

Once spatial data is constructed either by interpolation or remote sensing (or by both), thematic maps are generally derived. It is out of the scope of this paper to discuss classification and its accuracy in general, nevertheless, some notes should be made with regard to cartography.

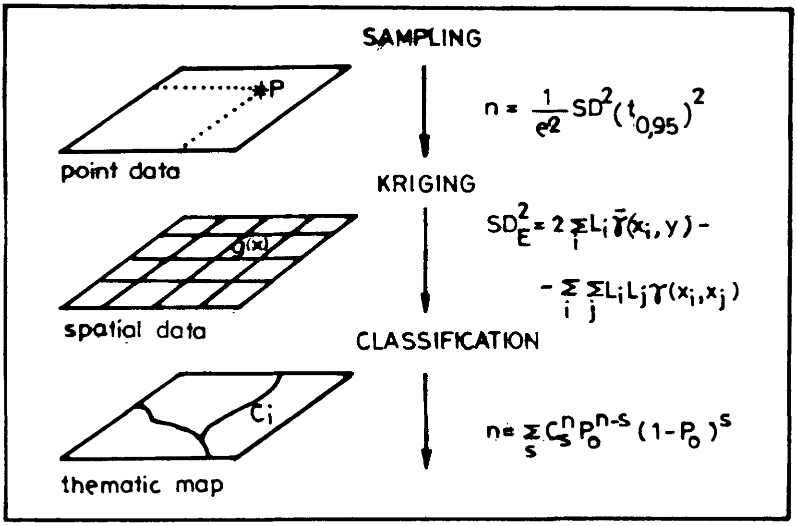

SAMPLING
P
point data
$n = \frac{1}{e^2} SD^2 (t_{0,95})^2$
KRIGING
g(x)
spatial data
$SD_E^2 = 2 \sum_i L_i \bar{\gamma}(x_i, y) - \sum_i \sum_j L_i L_j \gamma(x_i, x_j)$
CLASSIFICATION
C_i
thematic map
$n = \sum_s C_s^n P_0^{n-s} (1-P_0)^s$

Figure 2.

Error estimation at different stages of data processing (see text for details)

Thematic and geometric approach to the definiton of class boundaries

When some kind of preconditions (e.g. tradition) define the classes of a thematic map, it is only separability of the data set that effects recognition accuracy of such patterns. In case of a purely statistical data set, however, class intervals for a map should be determined with statistical considerations, such as to ensure the significant differences between adjacent classes beside the usual "equal-frequency" coding (Csillag 1986, Stegena and Csillag 1986).

There are some instances when the distribution of contours of a given thematic map is known. Burgess and Webster (1986), for example, have developed an algorithm incorporating a given distribution function to estimate the risk in constructing chloropleth maps by point sampling along transects.

Accuracy of thematic maps

When, as a supposed final product, a thematic map is constructed, its accuracy should be tested, too. Evaluating the accuracy of a thematic map requires sampling statistically the classified polygons to determine if the thematic classes, as mapped, agree with the field-identified categories.

Rosenfield et al.(1982) developed a method to validate the accuracy for each class with specified confidence from the cummulative binomial distribution. This method determines the minimum sample size for each category with a preliminary estimate of the accuracy.

AN EXPERIMENTAL SOIL MAPPING PROJECT

The above outlined ideas are being tested in an experimental soil mapping project in East-Hungary. Error propagation will be controlled in each step of processing from measurment error through interpolation to classification (see Fig.2). Soil samples have been collected at specific sites for calibration purposes and transect and gridded data have been collected for testing quantitative treatment of mapping errors. The project is envisioned as a pilot study to introduce this methodology in the TIR soil information system (Csillag et al. 1986).

A considerably large number of laboratory and in-situ calibration measurments have been completed to estimate the proportion of variance due to measurment errors.

Spatial patterns have been then described with semi-variograms (see Fig.3). These will serve as the basis for plotting spatial estimation error against aerial block size compatible with different sources of remotely sensed data, and for the interpolation of the training data with regard to spectral reflectance characteristics of soils (Baumgardner et al. 1985). Finally thematic maps are planned to be constructed and tested.

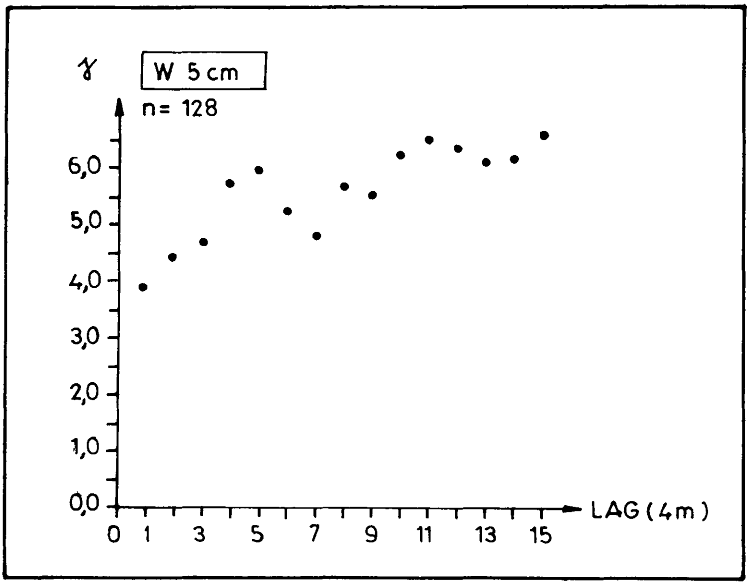


Figure 3.

Sample semi-variogram of soil moisture of the 0-5 cm layer

ACKNOWLEDGEMENT

The invaluable contribution to stimulating discussions of Sándor Kabos, Miklós Kertész and Kálmán Rajkai is warmly acknowledged. Many thanks to R. Webster for providing computer programs and consultation on geostatistics.

REFERENCES

Baumgardner,M.F., L.F.Silva, L.L.Biehl, E.R.Stoner (1985)
Reflectance Properties of Soils
Advances in Agronomy 38:1-44.

Burgess,T.M., R.Webster (1986)
A Computer Program for Evaluating Risks in Constructing Chloropleth Maps by Point Sampling Along Transects
Computers and Geosciences 12:107-127.

Burrough,P.A. (1986)
Five Reasons Why GISs Are Not Being Used Efficiently for Land Resource Assessment
in: AutoCarto London (ed:M.Blakemore) Vol.II. 139-148.

Carr,J.R., D.E.Myers (1984)
Application of the Theory of Regionalized Variables to the Spatial Analysis of Landsat Data
Proc. PECORA IX., IEEE Publ.CH-2079-2:55-61.

Csillag,F. (1986)
Comparison of Some Classification Methods on a Test-Site (Kiskòre, Hungary): Separability as a Measure of Accuracy
International Journal of Remote Sensing (in press)

Csillag,F.,S.Kabos,Gy.Várallyay,P.Zilahy,M.Vargha (1986)
TIR: A Computerized Cartographic Soil Information System
in: AutoCarto London (ed:M.Blakemore) Vol.II.

Curran,P.J., H.D.Williamson (1986a)
Sample Size for Ground and Remotely Sensed Data
Remote Sensing ofEnvironment 20:31-43.

Curran,P.J., H.D.Williamson (1986b)
Selecting Spatial Resolution for the Estimation of Grassland GLAI
in:Mapping from Modern Imagery, IAPRS 26:407-416.

Matheron,G. (1965)
Les Variables Regionalisées et Leur Estimation
Masson, Paris

McBratney,A.B., R.Webster, T.M. Burgess (1981)
The Design of Optimal Sampling Schemes for Local Estimation and Mapping of Regionalized Variables
Computers and Geosciences 7:331-334.

Meskó,A. (1984)
Digital Filtering: Applications in Geophysical Exploration
Akadémiai-Pitman-Halsted, Budapest-London-New York

Morrison,J. (1986)
Cartography: A Milestone and Its Future
in:AutoCarto London (ed:M.Blakemore) Vol.I. 1-13.

Oliver,M.A., R.Webster (1986)
Semi-Variograms for Modelling the Spatial Pattern of Landform and Soil Properties
Earth Surface Processes and Landforms 11:491-504.

Rosenfield,G.H., K.Fitzpatrick-Lins, H.S.Ling (1982)
Sampling for Thematic Map Accuracy Testing
Photogrammetric Engineering and Remote Sensing 48:131-137.

Stegena,L., F.Csillag (1985)
Statistical Determination of Class Intervals for Maps
(manuscript)

Webster,R. (1985)
Quantitative Spatial Analysis of Soil in the Field
Advances in Soil Science 3:1-70

December 16. 1986.

A MODEL OF ERROR FOR CHOROPLETH MAPS, WITH APPLICATIONS TO GEOGRAPHIC INFORMATION SYSTEMS

Michael F. Goodchild
and
Odette Dubuc
Department of Geography
The University of Western Ontario
London, Ontario, Canada N6A 5C2

ABSTRACT

The precision of geographic information systems is in sharp contrast to the accuracy of much spatial data, and requires a more objective approach than is conventional in cartography. Existing models of the error of cartographic lines are inappropriate for topological data for various reasons. We propose a model of error in choropleth data, with specific application to the data types found in natural resource inventories. One or more spatially autocorrelated continuous variables are generated, and mapped through a number of domains into a choropleth map with nominal attributes. Fractional Brownian surfaces are convenient sources of the continuous variables. The choropleth boundaries are subject to additional smoothing. Although the model is probably too complex to calibrate, it can be used to simulate choropleth images under a wide range of conditions, in order to investigate effects of error and accuracy in a variety of GIS functions.

INTRODUCTION

One of the more striking results of the introduction of digital data handling methods to cartography has been an increased interest in the interrelated issues of accuracy, precision, error and generalization. A digital system operates with a level of precision which is generally much higher than comparable manual methods, and often much higher than the accuracy of the data. For example, a point in a geographic information system might be represented by a pair of coordinates with a precision determined by the machine's floating point arithmetic, perhaps ten significant digits, whereas its location on a printed map might be accurate to no more than four digits, and might approximate a real feature on the ground to no more than three. The precision of various digital operations may also be far higher than is justified by the accuracy of the data or the conceptual basis of analysis. Poiker (1982, p.241) has compared the high precision of spatial data handling systems to "a person with the body of an athlete in his prime time and the mind of a child".

Statistical theory provides satisfactory methods for describing and dealing with error in scientific measurement, including surveying, but not to the same extent in cartography. Perkal's epsilon band (Perkal, 1956, 1966; Blakemore, 1984; Chrisman, 1982) has been used as an error model of cartographic lines in several recent studies (see also Honeycutt, 1986). Suppose there exists some abstract, true version of a line. Then the model proposes that all real representations of the line will lie within a band of error of width epsilon on either side of this true line. Blakemore (1984) has shown how this model can be used as the basis for a modified version of the point in polygon problem which explicitly recognizes the uncertainty in the location of a polygon boundary. Honeycutt (1986) has investigated the use of the model for distinguishing between spurious and real sliver polygons in topological overlay algorithms (see also Goodchild, 1978).

Despite the simplicity of the epsilon band concept, there are several reasons for believing that it is not completely satisfactory as a model of cartographic line error. First, although the model proposes that every line lies entirely within the epsilon band, we would expect intuitively that no such deterministic upper limit to error exists: instead, it would seem that larger errors are simply less likely. Error models of simple measurements, such as the Gaussian distribution, place no upper limit on the sizes of errors. Second, the model provides no distribution of error within the epsilon band. Although intuition might suggest that the most likely position for the real line is the centre of the epsilon band, in other words the true position, Honeycutt (1986) has found evidence that digitizing tends to produce a bimodal distribution, such that error on either side of the true line of some measureable amount less than epsilon is more likely than no error. These points suggest that a more suitable model would be some continuous distribution with asymptotic tails centred on the true line, the deterministic epsilon distance being replaced by a standard deviation parameter. The most suitable candidate would be a Gaussian distribution, or following Honeycutt (1986) an equal mixture of two Gaussians, one centred a distance to the left and one the same distance to the right.

Third, while the epsilon band and the modifications suggested above provide a model of deviation for a point on the line, they fail to model the line itself. The locations of two nearby points on the line are not chosen independently, but instead show a strong degree of autocorrelation. Furthermore, it is not clear which points on the line are modelled: Honeycutt (1986) analyzed the positions of digitized points, which are clearly not randomly and independently sampled from the set of all possible points on the line. So the error model cannot provide useful results about spurious sliver polygons, since these are formed not by the deviation of single points but by runs of autocorrelated points on both overlaid lines. A satisfactory error model would have to deal with the line as a continuum with strong autocorrelation.

The final objection to these methods concerns the nature of the data itself. Although it is convenient from a cartographic perspective to regard a line as an independently located feature with a true position, in reality many types of lines are subject to toplogical constraints, and are not independent of the areal features which they bound. For example a contour's position is not independent of other contours, since a large error in location may result in one contour crossing another. Contours are cartographic expressions of the value of some variable, often elevation, which is continuously distributed over the area. Problems with topological constraints on contour positions can be overcome if one regards error in contour position as an outcome of error in elevation, and concentrates on developing suitable models of elevation error instead. Fractional Brownian motion has been proposed as a suitable stochastic model of elevation (Mandelbrot, 1975, 1977, 1982; Goodchild, 1982; Goodchild et al., 1985; Mark and Aronson, 1984), in part because simulations using this stochastic process bear striking resemblance to some types of real terrain.

For the purposes of this discussion we can divide choropleth data into two types. The first, which we will refer to as socioeconomic, arises in fields such as the Census when a continuous variable is summarized using defined reporting units. The "cookie cutters" or unit boundaries are located in most cases independently of the variable being reported; in fact they may be used to report several hundred different and possibly unrelated variables. Error modelling is likely to be difficult since the process leading to error in each boundary depends on the nature of the boundary; lines which follow streets are likely to have very different errors from lines which are defined to follow rivers, for example. For this type of data it seems appropriate to separate error in attributes from error in feature location, as several authors have done (MacDougall, 1975; Chrisman, 1982), and to attempt to model each separately.

The boundaries of a choropleth map form an irregular tesselation of the plane. The literature contains a number of methods for generating random tesselations which might form useful models of error in choropleth boundaries (Boots, 1973; Getis and Boots, 1978; Miles, 1964, 1970). All of them satisfy the necessary topological and geometrical constraints on boundaries. However none are sufficiently irregular in appearance to be acceptable as simulations of real choropleth boundaries.

If a suitable method for generating boundaries could be found, the second stage of the simulation process would be to distribute attributes over the polygons in some reasonable fashion. A random allocation is unacceptable on two grounds; it fails to reproduce the spatial autocorrelation of attributes observed on almost all maps, and allows adjacent zones to receive the same attribute. Goodchild (1980) and Haining, Griffith and Bennett (1982) have discussed the simulation of autocorrelation.

The focus of this paper is on the other type of choropleth data, which we refer to as natural resource data. In this case boundaries are intimately related to the variable being mapped, and are in most cases unique to it. For example the boundaries on a soil map occur along lines of change in soil type, and are unlikely to coincide with boundaries on any other coverage. Boundaries are inherently uncertain, and the level of uncertainty is related to the change in soil class which occurs at the boundary; it is easy to believe that a transition from class A to class B might be more readily determined on the ground than a transition from A to C, for example. Under such circumstances it seems clear that an error model which separates attributes from locations must be inadequate.

The next section of the paper describes the proposed model. We then discuss the implications of the model for the analysis and description of natural resource data, and its potential applications.

THE MODEL

Consider a number *m* of continuous variables $z_1, z_2, \ldots, z_m$ distributed over the (x,y) plane. The variables will probably show spatial autocorrelation, and may or may not be correlated. Now consider an m-dimensional space defined by these variables; we will refer to this ts phase space by analogy to phase diagrams in thermodynamics. The space is divided into a number of domains, each of which is associated with one of a set of n classes:

$$C(z_1,z_2,\ldots,z_m) \in S \qquad (1)$$

where C is the class assigned to a point in phase space and S is the set of all possible classes. The domains provide a mapping from a set of m continuous variables to one of a set of classes. There may be more than one domain associated with a particular class, and some classes may not appear in the phase space. Finally, since the input variables are by definition continuous, it follows that if two zones share a common boundary on the choropleth map, then their corresponding classes must have been obtained from adjacent domains in phase space.

A simple model of world life zones by Holdridge et al. (1971) provides an illustration. Suppose that vegetation is largely controlled by temperature and precipitation variables, which have been mapped over the surface. Holdridge's diagram relating temperature and precipitation to vegetation class, reproduced in Figure 1, is a simple example of domains in phase space.

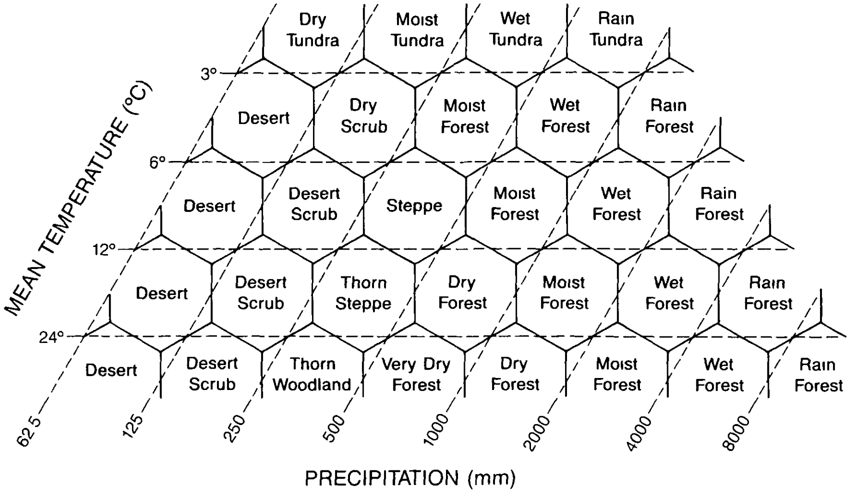


Figure 1. Example phase space for world life zone classification, from Holdridge et al. (1971).

If applied to the two input variables, it would map every combination of temperature and precipitation to a vegetation class, and thus convert two isopleth maps into one choropleth map. Errors in the choropleth map could then be ascribed to two sources: errors in the values of the continuous variables, and uncertainty in the delimitation of domains.

The visual appearance of the simulated choropleth map will clearly depend on the input surfaces. Highly irregular surfaces will produce highly fragmented choropleth zones, while smooth surfaces will produce large zones with relatively smooth boundaries, suggesting a direct relationship between the degree of spatial autocorrelation of the input surfaces and the nature of the resulting map. For this reason we propose to use fractional Brownian surfaces as input variables, because they allow control over the level of spatial autocorrelation: a single parameter H can be varied to generate a continuum from very smooth (H=1) to very rugged (H=0) surfaces. A value of 0.7 has often been identified as giving the closest visual appearance to real terrain (Mandelbrot, 1977, 1982).

To illustrate the model, two surfaces were generated, at H=0.7 and H=0.6, and sampled with a 64 by 64 array. Each cell's values of z_1 and z_2 were mapped into the five-class phase space shown in Figure 2: the 4096 points are shown as dots. The resulting classified raster was vectorized to give the polygons shown in Figure 3.

It is likely that the boundaries produced by this simulation process are too irregular to be acceptable: they also show many isolated islands, which are rare on real maps. We suggest that these differences are the result of cartographic smoothings which take place during the drawing of choropleth boundaries. To allow for this, and also to remove the visual effects of pixel boundaries, we have added

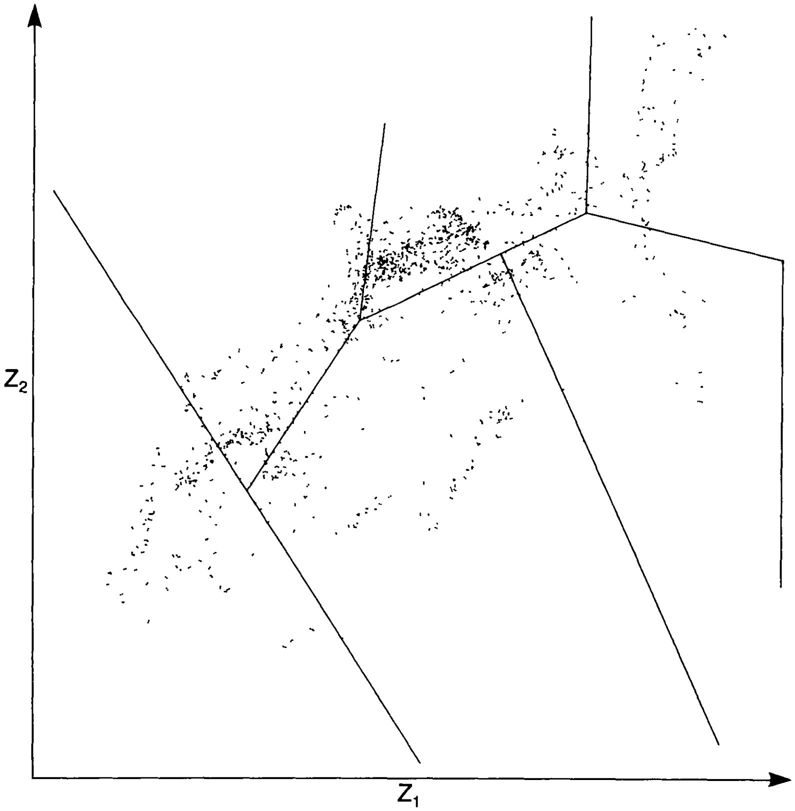


Figure 2. Phase space used in example simulation, with points from two 64 by 64 rasters.

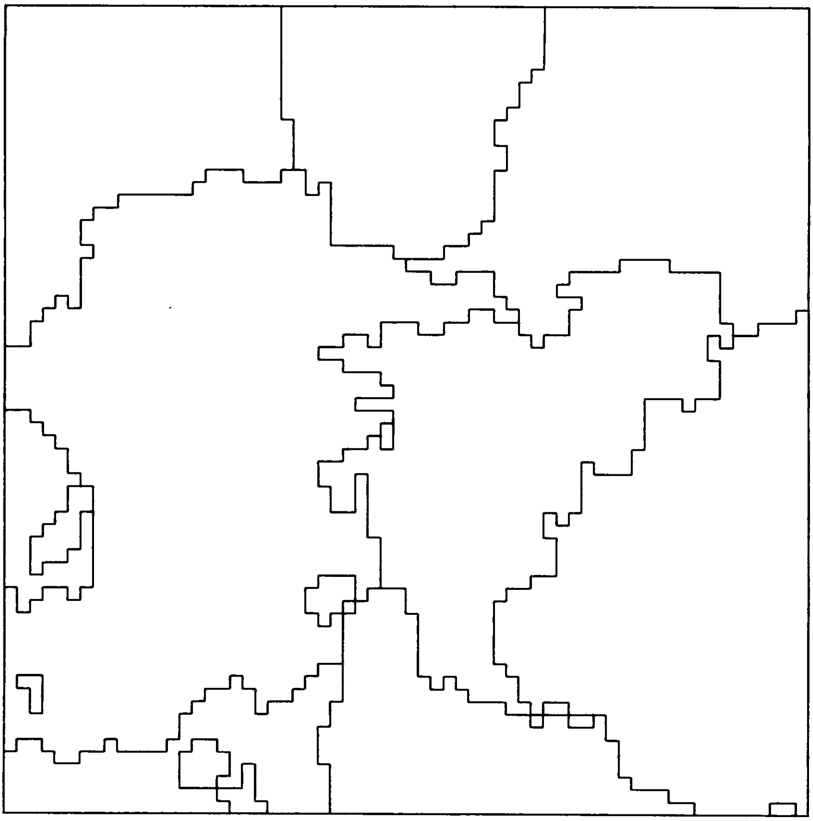


Figure 3. Classified 64 by 64 raster simulation.

two stages to the simulation process. First, the vectorization algorithm has been biassed against small islands. The normal criterion for contiguity is rook's case: a cell is not part of a larger choropleth zone unless at least one of its four rook's case neighbours is also part of the zone. However we allow an additional case: a pixel can be part of a larger zone if at least one of its bishop's case (diagonal) neighbours is also part of the zone, provided all of its four rook's case neighbours are part of some other, second zone. Second, we smooth the vectorized boundary between topological vertices by using a simple spline. This has the effect both of removing the pixel outlines, and also of reducing the irregularity of the line to emulate the cartographer's implicit generalization.

IMPLICATIONS OF THE MODEL

A contour map can be seen as a choropleth map in which the zones between every pair of adjacent contours are given a unique colour or class. In terms of our model, this choropleth map would be generated from a single variable, m=1, using a phase space of one dimension in which the domains appear as divisions along the axis of that variable. Classes can be adjacent on the choropleth map, and have non-zero common boundary length, only if their corresponding domains are adjacent in phase space. It follows that there is a unique ordering of the classes such that when adjacencies are counted in a table in which the classes have been placed in the correct order in both rows and columns, the only non-empty cells will be those immediately adjacent to the diagonal.

The same property holds for two input variables if the domains are bounded by parallel lines, and similarly for more than two variables. If domain boundaries are parallel, it follows that some linear combination of the two input variables can be found, perpendicular to the domain boundaries, which would produce the same choropleth zones.

In terms of the model, the relative frequencies of adjacencies on a choropleth map are therefore an indication of the complexity of the phase space and the number of input variables, independently of the error or distortion of the data. For example, error can never produce an adjacency between two classes which are not adjacent in the underlying phase space. It can, however, produce an adjacency which was not previously present on the choropleth map but which is nevertheless present in phase space.

While the model replicates the observed crude characteristics of much natural resource choropleth data, we do not wish to imply that all such data is generated by processes of this type. The model seems reasonable as a mechanism for determining vegetation zones in relation to continuous, climatological variables, but no comparable continuous variables control bedrock geology or soil class. Some characteristics of choropleth data are clearly not replicated, such as the long, contorted polygons which follow rivers on maps of floodplains and related phenomena.

APPLICATIONS

The model provides a method for simulating choropleth boundary networks and associated attributes under a variety of conditions from small, fragmented zones to large ones and from highly irregular boundaries to smooth ones. We plan to use it to investigate a number of questions related to error and accuracy in choropleth maps, the answers to which are significant in the design and operation of geographic information systems.

First, the model will allow us to investigate the relationships between the accuracy of a spatial data base and the accuracy of measures derived from it, under a full range of conditions. For example, there is need for empirical work to examine further the effects of pixel size in raster data bases, and of digitizing errors and line generalization in vector data bases. The use of simulated rather than real data allows greater control over the characteristics of the data, and a wider range of experimental conditions.

Second, the model may provide a better understanding of the sources of error in choropleth maps. Error can occur at several stages in the simulation; in the measurement of the continuous variables, in the spatial sampling design (the density and position of the raster), in the delimitation of domains in phase space, and in the vectorization and smoothing of polygon boundaries. Each of the various sources of uncertainty in a soil map boundary can be related to one or more of these sources. For example, uncertainty may be due to a low density of sampling of soils near the boundary, to inaccurate measurement of parameters such as soil colour, to subjective smoothing of the boundary by a cartographer, or to imprecision in the definition of soil classes. It is possible to simulate each of these separately, and to observe their effects. For example, error due to smoothing will produce uniform uncertainty for all lines, whereas error due to inaccurate measurement of one underlying continuous variable will produce degrees of error in boundary lines which are a function of the classes separated by the line, and depend directly on the slope of the relevant domain boundary in phase space.

Third, we can observe the effects of each source of error on GIS operations such as polygon overlay and sliver removal. Algorithms designed to remove slivers can be tested under a variety of conditions and forms of error.

One of the more desirable objectives of a study of error in spatial data bases would be the development of hypothesis tests to resolve such questions as whether a particular sliver polygon is real or spurious, based on its area or shape, or whether a point lies inside or outside a polygon. To do so would require a simple error model characterized by a very small number of parameters. The model proposed here is clearly not suitable; its parameters include the number and level of spatial autocorrelation of the underlying continuous variables, the spatial sampling design, the geometry of the phase space and the nature of the splining process. Although various simplifying assumptions might be made (for example that all boundaries in phase space are straight), there seems little prospect of calibrating a model of this complexity.

Greenland and Socher (1985) have proposed a simple measure of the degree of agreement between two versions of the same choropleth map. The proportion of area which has been assigned the same class on both maps, p_o, is compared to an expected proportion p_e in an index denoted by kappa. The basis for the calculation of p_e is the assumption that class is randomly allocated, in other words that the proportion of area allocated to class A on one map and to class B on the other is simply the product of the proportion which is A on the first map and the proportion which is B on the second.

If the maps show highly fragmented polygons, it is relatively easy for errors in boundary positions to produce agreements no higher than the expected proportion, and thus low values of kappa. But if the polygons are large, the same degree of boundary distortion will reduce kappa only slightly, and it will be almost impossible to find distortions which yield low kappa values. In other words, kappa is highly sensitive to the degree of spatial autocorrelation in attributes, and cannot be compared usefully across different types of data. To do so requires a more appropriate model of error. However given the variety of possible sources and forms of error in the model proposed in this paper, it is unlikely that a simple measure of data base distortion could be devised which would be valid across a range of data types.

REFERENCES

Blakemore, M. 1984, Generalization and error in spatial databases: Cartographica 21, (2,3), 131-139.

Boots, B.N. 1973, Some models of the random subdivision of space: Geografiska Annaler 55B, 34-48.

Chrisman, N. 1982, Methods of spatial analysis based on errors in categorical maps: Unpublished Ph.D. thesis, University of Bristol.

Getis, A. and Boots, B. 1978. Models of Spatial Processes, Cambridge University Press, London.

Goodchild, M.F. 1978, Statistical aspects of the polygon overlay problem: Harvard Papers on Geographic Information Systems 6, Addison-Wesley, Reading, Mass.

Goodchild, M.F. 1980, Simulation of autocorrelation for aggregte data: Environment and Planning A 12, 1073-1081.

Goodchild, M.F. 1982, The fractional Brownian process as a terrain simulation model: Modelling and Simulation 13, Proceedings of the 13th Annual Pittsburgh Conference, 1133-1137.

Goodchild, M.F., Klinkenberg, B., Glieca, M. and Hasan, M. 1985, Statistics of hydrologic networks on fractional Brownian surfaces: Modelling and Simulation 16, Proceedings of the 16th Annual Pittsburgh Conference, 317-323.

Greenland, A. and Socher, R.M. 1985, Statistical evaluation of accuracy for digital cartographic bases: Proceedings, AutoCarto 7, 212-221.

Haining, R.P., Griffith, D.A. and Bennett, R.J. 1983, Simulating two-dimensional autocorrelated surfaces: Geographical Analysis 15, 247-253.

Holdridge, L.R., Grenke, W.C., Hathaway, W.H., Liang, T. and Tosi, J.A. Jr. 1971. Forest Environments in Tropical Life Zones: A Pilot Study, Pergamon, Oxford.

Honeycutt, D.M. 1986. Epsilon, generalization and probability in spatial data bases, Unpublished manuscript.

MacDougall, E.B. 1975, The accuracy of map overlays: Landscape Planning 2, 23-30.

Mandelbrot, B.B. 1975, Stochastic models of the Earth's relief, the shape and the fractal dimension of the coastlines, and the number-area rule for islands: Proceedings of the National Academcy of Sciences 72, 3825-3828.

Mandelbrot, B.B. 1977. Fractals: Form, Chance and Dimension, Freeman, San Francisco.

Mandelbrot, B.B. 1982. The Fractal Geometry of Nature, Freeman, San Francisco.

Mark, D.M. and Aronson, P.B. 1984, Scale-dependent fractal dimensions of topographic surfaces: Mathematical Geology 16, 671-684.

Miles, R.E. 1964, Random polygons determined by random lines in a plane: Proceedings of the National Academy of Sciences 52, 901-907, 1157-1160.

Miles, R.E. 1970, On the homogeneous planar Poisson point process: Mathematical Biosciences 6, 85-127.

Perkal, J. 1956, On epsilon length: Bulletin de l'Academie Polonaise des Sciences 4, 399-403.

Perkal, J. 1966, On the length of empirical curves: Discussion Paper 10, Michigan Inter-University Community of Mathematical Geographers, Ann Arbor.

Poiker, T.K. 1982, Looking at computer cartography: GeoJournal 6, 241-249.

UNCERTAINTIES IN LAND INFORMATION SYSTEMS DATABASES

Yvan Bédard, Ph.D.
Université Laval
Département des Sciences géodésiques et de Télédétection
Pavillon Casault, #1344
Québec, Qc
Canada G1K 7P4

ABSTRACT

Based on the communication paradigm of Land Information Systems, this paper presents (1) how uncertainty is inevitably introduced in LIS databases, (2) four resulting types of uncertainty, and (3) different means to deal with uncertainty. Finally, the paper suggests that there exist two classes of land data with regards to their reliability.

INTRODUCTION

Land Information Systems (LIS) are useful only to the extent that the information they provide effectively reflect the Reality. However, several limitations affect the veracity of the data stored in spatial databases and concerns about their reliability are highlighted in recent literature (see for example Bouillé 1982; Craig 1983; Blakemore 1983; Dutton 1984; Dangermond, Derrenbacher and Harnden 1984; Robinson and Strahler 1984; Robinson and Frank 1985; Zwart 1985; Bédard 1986 a-b). The following pages relate to these concerns, they present (1) the causes affecting the veracity of land data, (2) four resulting orders of uncertainties, and (3) different solutions to better deal with these uncertainties.

The paper is based on Communication Sciences, Information Theory and Computer Sciences concepts. The overall analysis is built upon the "communication paradigm of LIS" as described by Bédard (1986 a-b) where (1) the terms "data" and "information" are respectively used for physical and cognitive models, and (2) a LIS is seen as an indirect sequential communication process between collectors and users of data.

THE COMMUNICATION PARADIGM OF LAND INFORMATION SYSTEMS: AN OVERVIEW

According to the general framework introduced by Bédard (1986 a-b), Land Information Systems begin with observations of the *real world* which is assumed objective, independent of the observers and which is very complex.

To make decisions about this world, *abstraction* is necessary. To do so, humans selectively perceive the reality where living beings, objects, places, events, or their surrogates emit or reflect different *signals* (light, sounds, odors, etc.). LIS observers, like land surveyors, pick up those signals through their five senses which are sometimes assisted by technical extensions such as amplifiers and translaters.

The detected signals travel to the observer's brain to be *recognized.* This consists of matching the detected signals with previously stored "referents" in order to give meaning to those signals. Then, the observer mentally reconstructs the observed part of the reality. This *cognitive image* is the first model of the observed reality in the LIS communication process. It is assumed partly subjective since it depends not only on the reality, but also on the observer and the context.

Afterwards, the observer must *communicate* his cognitive model to the LIS central agency and the LIS users. However, mental models cannot be communicated directly; physical counterparts must be created like sounds, drawings, and writings. Using the right physical counterpart to communicate the desired meaning is called *encoding;* it is the processus used to create data. Those encoded signals, when put together, form a second model of the reality. They form a *physical* model ready for communication. However, this physical model stems from the observer's perception of the reality, not the reality.

Those encoded signals, which are transmitted, have two components: a *content* and a *form.* When we communicate, only one component is transmitted: the form. Nevertheless, those signals convey an *intended* content (which is not built in the data).

Those signals or land data are usually transmitted to an intermediary, an LIS central agency which performs several gatekeeping functions such as implementing modelization and communication rules and checking for data quality. This central agency usually stores the transmitted data in its database, i.e. the LIS database.

Afterwards, copies of the stored land data or new ones created from those previous ones are sent to the LIS users. Those physical signals reach the receiver's sensory organs (or technical extensions) and travel to his brain to be *decoded;* that is, the receiver must perform the inverse operation of the sender's encoding, but with his *own* referents. He must "guess", among potential meanings, the one conveyed by the received signals. To do so, he must know the context and the language used by the encoder because the same content may have different forms and the same form lead to different contents. This interpretation of the message symbolically sent to him allows for the creation of his own cognitive model of a part of the reality, a part that he has probably *not* observed himself. This is the fundamental basis of the communication paradigm of LIS.

It is only when this latter model of the world, created by LIS data users from physical models instead of the reality, is made that the LIS communication process is completed. Thus, a LIS is a sequence of cognitive and physical modeling processes (where the number of model increases with the number of intermediaries in the LIS communication process) (see figure 1).

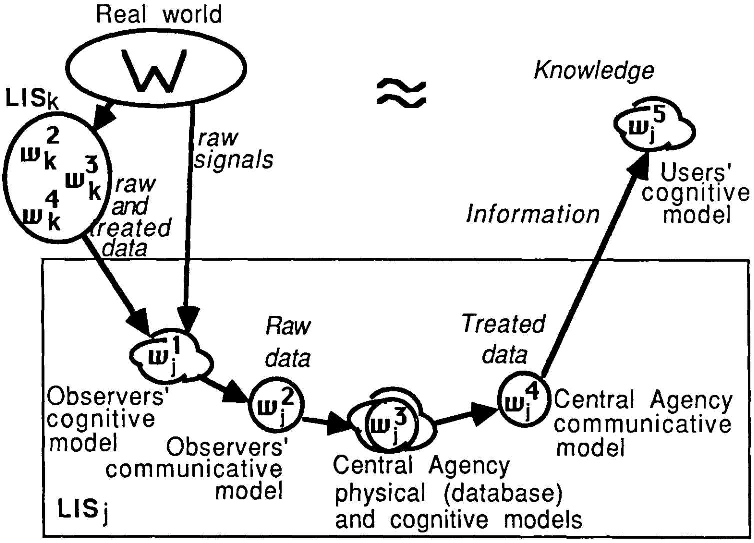


Figure 1: General sequence of model buildings in LIS (from Bédard 1986a) (circles and clouds respectively represent physical and cognitive models).

UNCERTAINTY IN LAND INFORMATION SYSTEMS DATABASES: THE CAUSES

Because uncertainty is introduced each time a model is built, the LIS database w^3 cannot be a perfect model of the reality. As stated by Bouillé (1982), "cartographic results are fuzzy and have been since the earliest beginning of cartography"; such a point of view is endorsed by Robinson and Strahler (1984) who also wrote that "GIS's should be thought of as containing inexact representations of segments of reality".

The uncertainty introduced each time a model is built comes from *two sources*. *First*, it comes from the intrinsic limitations of the modelization process itself; *second*, it comes from the model-makers.

Limitations inherent to the modelization process itself

There are several limitations which are intrinsic to the modelization process, for example:

1- loss of details,
2- purposefulness or goal dependency,
3- model-maker dependency,
4- context dependency,
5- translations between cognitive and physical models are not straightforward,
6- modelization and communication rules are rarely unequivocal.

However, the most important limitation probably is that models are only approximative estimations. This is explained by two kinds of *estimation limitations*: (1) fuzziness in the identification or labeling of an entity, and (2) limitations in the measurement of the properties of those entities.

1- Fuzziness in identification takes place when humans classify realities into discrete groups of entities that do not have sharp boundaries. This can happen (1) on the boundary of an entity type between existing and not existing (e.g. when to attach the label "tree" to a tree? the label "fence" to a fence? "wetland" to a wetland?), and (2) on the border between two types of entities (e.g. in the Quebec assessment file, a multipurpose building which can be classified as entity types "commerce" or "industry" will be described by different attributes depending upon the final choice).

The problem is that the world generally is a continuum while labels are discrete; however, "for many aspects of the world, a discrete set of concepts is adequate.... Yet such distinctions break down when pushed to extremes" (Sowa 1984, 344). The mistake is to suppose that realities are clear-cut and self-identifying. On the contrary, many of our perceptions involve the ambiguous and the intangible. This fuzziness in the identification introduces uncertainty in the existence of an entity or in its classification in the good entity type.

2- Limitations in the measurement of the properties of the observed entities are related to its qualitative or quantitative character (Robinson and Frank 1985). Those limitations are of two kinds: (1) *imprecision* (in its mathematical sense) which is related to the quantitative measurement of attribute values (e.g. standard deviation of 3 cm) and (2) *fuzziness in the qualitative measurement* of attribute values where classes of values do not have sharply defined boundaries (e.g. the building quality codes used in Quebec assessment files: mauvais état, médiocre-, médiocre, standard-, standard, standard+, bon, bon+, excellent état). This fuzziness is in fact the same phenomena than fuzziness in identification but applied at a different level of abstraction. Those limitations in measurement introduce uncertainty in the attribute values stored in a database.

Thus, there are two kinds of estimation limitations which affect LIS users in "knòwing what they talk about" (fuzziness in identification) and "describing and locating it" (measurement limitations). Those estimation limitations have different consequences in the content of a database: while fuzziness in identification affects the type of an entity and consequently which properties are measured for this entity, limitations in measurement affect the values of the measured properties. Thus, a fuzzy identification affects the reliability of the entity as a whole while limitations in measurement individually affect the attribute values of an entity.

Limitations related to model-makers

Land data uncertainty is also related to the model-makers involved in the LIS communication process; i.e., the collectors, intermediaries and users of LIS data.

As we can deduce from figure 1, a large amount of LIS data involve human judgments. However, there is a gap between humans' imprecise knowledge of the reality and the crisp representations of the reality which are stored in LIS databases.

Humans, as information processors, have limited capabilities and introduce subjectivity in data. It is commonly accepted that even in the best conditions, a same reality will *not* be modeled the same way by different persons or by the same person in different times and contexts. As stated by Dutton (1984), "our models of reality, including cartographic databases, are highly conditioned by our cultural and institutional consensus concerning 'what exists'. But that consensus varies across space, differs among groups, shifts over time, and is colored by our concerns". Also, "there is an implicit assumption that the information contained in land information systems is objective, quantified and correct (i.e. scientific or rational data)... the data contained within land information systems is neither totally objective nor necessarily based on acceptable statistical measures" (Zwart 1985).

Communication scientists explain this phenomena by the influence of everyone's *frame of reference* particularities, meaning the influence of someone's history, experience, learning, needs, aspirations, beliefs, values, and personality. This also includes someone's group norms: cultural, professional, and familial.

Also responsible is the *concept of "satisficing"* where someone does not automatically analyse all the possibilities to select the best representation of the real world. Instead, he limits his search for the best solution by accepting the first alternative satisfying all the given requirements (Davis and Olson 1985, 169). This method is very frequent in LIS activities (e.g. differences in cost and time between Quebec subdivisions and "bornages", two operations delimiting rights to the land but with only the latter one having a legal value).

Everyone's frame of reference and use of the satisficing concept influence directly the reliability of LIS databases. This influence may happen during the perception (detection and recognition) of the reality raw signals, during the perception (detection and decoding) of raw and treated data, during the creation of cognitive models, and during the encoding of cognitive models into physical models (e.g. LIS databases).

Thus, uncertainties stemming from the modelization process itself and the model-makers are unavoidable. Consequently, there is an inherent uncertainty in land data which cannot be avoided and LIS cannot deliver perfect information. At best, LIS databases can only be workable approximations of the real world W.

THE PROBLEM

Land information systems communicate models of parts of the real world to identify land-related entities, to describe them and to locate them in space and time. However, even in the best conditions, there are uncertainties affecting the reliability of LIS databases. To better understand the consequences of those uncertainties, i.e. the resulting problem, the following classification has been done:

1- First order *(conceptual)* uncertainty: refers to the fuzziness in the identification of an observed reality (e.g. being or not being such an entity? Being an entity of type A or of type B?).

2- Second order *(descriptive)* uncertainty: refers to the uncertainty in the attribute values of an observed reality (i.e. imprecision in quantitative values and fuzziness in qualitative values).

3- Third order *(locational)* uncertainty: refers to the fuzziness in the qualitative values and imprecision in the quantitative values used for the location in space and time of an observed reality (e.g. error ellipses in geodesy).

4- Fourth order uncertainty *(meta-uncertainty)*: refers to the degree to which the preceding uncertainties are unknown (e.g. absolute error ellipses with a probability of 39.3%; being pretty sure that a building quality is "standard +").

Those four orders of uncertainty combine to each other to generate the total uncertainty in LIS databases, leading to an uncertain information about the real world. The result is a user who doubts if a given reality is in fact such an entity in the real world, if it really has the given attribute values, if it really is where it is depicted, and if the level of those uncertainties is high or low.

THE SOLUTIONS

As previously seen, there is an inherent uncertainty in LIS databases which cannot be avoided. We can take means to (1) reduce this uncertainty and to (2) absorb partially or completely the remaining uncertainty. The right balance among those alternatives depends upon political, cultural, and economical concerns; it is an institutional choice to be done within each jurisdiction.

Uncertainty reduction

Uncertainty reduction takes place when modelisation rules (defining the content of a model, i.e. what to observe and how) and communication rules (defining the form of a model, i.e. which graphical and literal languages to use) are established either (1) to decrease the fuzziness associated with the identification of a spatial entity or (2) to insure precision and crispness in the description and location in space and time of this spatial entity.

This can be done by appropriate technical, procedural, organizational and legal requirements such as geodetic tying of surveys, use of mathematics such as adjustments for repetitive quantitative measurements and fuzzy logic for qualitative measurements, good professional training, high precision standards, mandatory marking of property corners, use of standard symbols, inclusion of lineage in digital maps, mandatory registration of all the rights to the land, etc. Such methods increase the likelihood that the several models which are built in the LIS communication process will correspond more closely to the observed reality.

Any LIS reduces the uncertainty inherent in land data to a certain degree. However, this is limited by fundamental concepts as well as practical and economical conditions. Furthermore, although we can reduce the uncertainty inherent in land data, we cannot eliminate all of it. Thus, there remains, in the LIS communication process, someone who absorbs, in whole or in part, the effects of the remaining uncertainty.

Uncertainty absorption

Uncertainty absorption takes place when a model maker guarantees his model of the reality and compensates the users damaged by poor data. Uncertainty absorption also takes place when non-guaranteed models are utilized. Here, the user and not the provider of data absorbs the uncertainty. In fact, the level of uncertainty absorption is defined as the level of (monetary) risk in providing or using data. When errors in data cause damages to users, the ones who pay for these damages are the ones who absorb the uncertainty.

Uncertainty absorption is very different from uncertainty reduction. In the latter case, the uncertainty is literally reduced (ex. requiring a precision of 3cm instead of 10cm, asking for the opinion of two or three land surveyors instead of only one). In the former case however, there is someone who guarantees the data as the "truth" and who is willing to take the inherent risk (e.g. guarantee of titles and boundaries with indemnity funds in the Massachusetts registration system).

Only the absorption performed or made official by the LIS central agency (or its extension like a tribunal) applies to all the participants in the LIS communication process. In such cases, the LIS central agency has the power and means to impose a specific model of the world as the "good one". When this happens, an LIS database (or part of it) becomes the "official" view of the reality, a kind of "artificial truth" binding every participant in the LIS. Although these models do not necessarily represent exactly the reality, they represent the "official" version, the "official" model of this reality and they are guaranteed.

Such an alternative almost eliminates the uncertainty inherent in the original nature of data. In fact, it really absorbs the remaining uncertainty and decisionmakers can better rely on such data. Users of those data must and can rely on them.

It is interesting to note at this point that most of the ways to reduce uncertainty are technical, while most of the ways to absorb the remaining uncertainty are institutional. Finally, the higher the uncertainty reduction, the lower the uncertainty absorption needed.

CONCLUSION

Land data are physical and formal symbolic surrogates created by humans to communicate information about the description and location of land-related realities. Thus, LIS databases contain the symbols of the LIS communication process with their inherent uncertainty. However, the four resulting orders of uncertainty can be reduced and absorbed by appropriate means. This gives rise to two types of data in LIS databases:

a) *Second class* land data: this is the typical data found in LIS databases, they have a certain degree of remaining uncertainty which has not been absorbed. These data are approximative surrogates which lie on a spectrum going from *vague* on one end (data with a lot of uncertainty remaining) to *exact* on the other end (little uncertainty remains).

b) *First Class* land data: this type of land data is rare in LIS databases, this is the data for which the uncertainty has been absorbed. These land data are the *official model* binding every participant in the LIS. The original nature of these land data has been changed to "artificial truth" and no uncertainty remains for them.

Exact land data are very good approximations of the reality. But, only the First Class land data can be considered as having a complete reliability.

CITED REFERENCES

Bédard, Y. 1986a, A Study of the Nature of Data Using a Communication-Based Conceptual Framework of Land Information Systems. Ph.D. Dissertation, University of Maine, Orono, 260 pp.

Bédard, Y. 1986b, A Study of the Nature of Data Using a Communication-Based Conceptual Framework of Land Information Sytems, Updated Version. First presented at the XVIII Congress, Fédération Internationale des Géomètres, Toronto, June 1-11, Volume 3. Updated version reprinted in The Canadian Surveyor, Winter Issue.

Bédard, Y. 1986c, Comparing Registration Systems Using a Communication-Based Criterion: Uncertainty Absorption. XVIII Congress, Fédération Internationale des Géomètres, Toronto, June 1-11, Volume 7.

Blakemore, M. 1983, Generalization and Error in Spatial Data Bases. Proceedings of the Sixth International Symposium on Automated Cartography, Ottawa, October 16-21, Volume 1, pp. 313-322.

Bouillé, F. 1982, Actual Tools for Cartography Today. Cartographica, Volume 19, No.2, pp. 27-32.

Craig, W.J. 1983, Problems of Spatial Accuracy. Decision Support Systems for Policy and Management. Annual Conference of the Urban and Regional Information Systems Association, August 14-17, Atlanta, pp. 1-10.

Dangermond, J., B. Derrenbacher, and E. Harnder 1984, Description of Techniques for Automation of Regional Natural Resources Inventories. Appendix 2: Error Potential Related to Integrated and Parametric Mapping in GIS. In Seminar on the Multipurpose Cadastre: Modernizing Land Information Systems in North America. B.J. Niemann Ed. Madison, Wisconsin: the University.

Davis, G. B. and M. H. Olson 1985, Management Information Systems: Conceptual Foundations, Structure, and Development. 2nd ed. New York: McGraw-Hill.

Dutton, G. 1984, Truth and its Consequences in Digital Cartography. Technical Papers, the American Congress on Surveying and Mapping, 44th Annual Meeting, Washington, D.C., March 11-24, pp. 273-283.

Minsky, M. L. 1965, Matter, Minds, and Models. Proceedings of the International Federation of Information Processing Congress, Vol. 1, pp. 45-49. Reprinted in Marvin L. Minsky, ed. Semantic Information Processing. MIT Press, 1968.

Robinson, V. B. and A.H. Strahler 1984, Issues in Designing Geographic Information Systems under Conditions of Inexactness. Tenth International Symposium, Machine Processing of Remotely Sensed Data, West Lafayette, Indiana, June 12-14, pp. 198-204.

Robinson, V. B. and A. U. Frank 1985, About Different Kinds of Uncertainty in Collections of Spatial Data. Proceedings of the Seventh International Symposium on Automated Mapping: Digital Representations of Spatial Knowledge. American Society of Photogrammetry and American Congress on Surveying and Mapping. Washington, D.C. March 11-14, pp. 440-449.

Sowa, J. F. 1984Conceptual Structures: Information Processing in Mind and Machine. Reading, Mass.: Addison-Westley.

Zwart, P. 1985, Response Paper on Assessing User Requirements and System Design Approaches to Land Information Systems. Workshop on Fundamental Research Needs in Surveying, Mapping, and Land Information Systems. November 17-20, Virginia Polytechnic Institute and State University.

SPATIAL ORDERING OF VORONOI NETWORKS AND THEIR USE IN TERRAIN DATA BASE MANAGEMENT

Christopher M. Gold

College of Geographic Sciences,
P.O. Box 10, Lawrencetown
N.S., Canada BOS 1MO
BITNET address: COGS@ACADIA

ABSTRACT

"Computational Geometry" traditionally included topics such as curve and surface fitting; more recently it has been concerned with minimizing computational complexity in a global sense. Another aspect of spatial problems still needing attention is the concept of spatial ordering of data so as to minimize data base access, pen movement, etc., as well as to answer questions concerning the adjacency of objects on a map.

If we take point data for terrain modelling as our illustrative problem, and create a Delaunay triangulation as our spatial data base, we may define a limited set of authorized transactions that may access the data base. These include search, insert and delete of data points, and radial search outwards from some initial viewpoint. These transactions (and especially the last) are particularly critical for systematic map accretion procedures and field-of-view problems. Based on these data base considerations maps, block diagrams and field-of-view calculations may be generated in a consistent fashion from similarly-constructed data bases, and a generic procedure for producing block diagrams is given as an illustration of the approach.

INTRODUCTION

While the term "computational geometry" has traditionally included such topics as curve and surface modelling (Faux and Pratt, 1979), more recently the emphasis appears to have been leaning towards a more formal role, e.g. in defining the complexity of a particular geometric problem (e.g. Preparata and Shamos, 1985). Other perspectives may, however, still fall within the broader definition of the term. One of these is the concept of spatial ordering of data - in this case in two dimensions so that individual data points, etc., may be referenced in a consistent and predictable fashion, despite the predilection of traditional computers for handling information in a sequential linear fashion.

This topic is relevant to the computer-based mapping business, since technology is generating ever-more detailed data, covering extremely large areas, and demanding increasingly rapid updating of the appropriate data-base, often only modifying a few local areas at one time. While regional questions, for example the merging of two adjacent data sets, are of great importance, it is becoming increasingly necessary to be able to operate effectively within a very small part of a very large cosmos, without perturbing unnecessarily those regions not relevant to the immediate local issue.

EFFICIENCY OF NETWORK OPERATIONS

Following from these considerations is the philosophy that: A) individual operations on the data (whether interrogation or modification) are local in nature; and B) subsequent operations tend to be near previous ones. This affects significantly the appropriate definition of computational efficiency - in general, operations may be O(n) once the neighbourhood or relevant portion of the data base has been reached. Getting there, however, may be significantly less than half the battle - if the previous operation was nearby. Thus search procedures may sometimes be tolerated that are of less than the theoretical maximum order of efficiency. Preparata and Shamos (1985) indicate that in two dimensions, point Voronoi diagrams may be constructed in O(n log n) time. Experience with triangular networks indicates that operations are O(n) (i.e. local in nature) except for the search through the data structure to find the appropriate local element. Direct walk methods (e.g. Gold et al., 1977) are O(n**1.5) in theory, but in practice they rarely match the worst case. Improved data structures could reduce the search time to O(n log n) if necessary.

An earlier publication (Gold, 1984) examined the problem of terrain modelling or contouring from arbitrarily distributed data points. He broke the problem into five stages: data point entry and retrieval; sample site selection for surface estimation; neighbouring point selection; surface estimation procedures; and display methods. Apart from noting that surface estimation (interpolation) is heavily dependant on neighbouring point selection, our interest in these steps concerns the relation between neighbour selection and data entry or retrieval - in other words, effective utilization of an appropriate data base. For reasons mentioned previously the Voronoi tesselation appears to be a good general purpose measure of neighbourhood relationships. Gold et al.(1977) described a triangulation based data structure for terrain modelling. Lawson (1977) has described a criterion that is equivalent to Delaunay triangulation. Both workers used a technique of switching the diagonals of a quadrilateral formed by two adjacent triangles in order to improve the triangulation. An interesting sidelight on this approach is that using the optimization criterion of Gold et al.(1977), which does not produce a global optimum, approximately 5.7 diagonal switches were required to (locally) optimize one data point, ignoring boundary conditions. For the Lawson case, which is equivalent to the (global) Delaunay criterion (i.e. the triangulation which is the dual graph of the Voronoi tesselation) precisely 6 switches were needed on the average. An excellent summary of the computational geometry approach was given by O'Rourke (1984). It is the earlier work that interests us here - in particular the view of a Delaunay triangulation as a data base.

DATA BASES AND SPATIALLY ORDERED TRANSACTIONS

Perhaps the most convenient form for preserving the triangulation is as a file with one record per triangle, containing pointers to each adjacent triangle and each adjacent data point (vertex). This is conceptually convenient as it separates objects from their spatial relationships. If the object and relationship files (or tables) are to be treated as a data base, only certain authorized transactions may be performed on the underlying triangulation. Based on previous work (Gold et al., 1977, Gold and Maydell, 1978, Gold, 1984) some authorized transactions are:

"Location Search", which walks through the network to find the enclosing triangle for a specified coordinate location;

"Insert", which sub-divides the enclosing triangle into three, therefore updating the relational linkages to accommodate a new data point;

"Switch" or "Optimize" which adjusts individual triangle pairs until the Delaunay (or other) criterion was achieved in the neighbourhood of the new data point;

"Rotational Search" which retrieves the neighbouring points or triangles to an already-entered data point;

"Radial Search" which, given some central reference location, scans outwards from this, retrieving all points, triangles or edges until some terminating criterion is achieved.

Locational Search, Insert and Switch are basic operations required for network generation, and have been described previously. Rotational Search is used when the immediate neighbours are required to some point in the data base - for example, for estimating its slope coefficients. The Radial Search procedure (Gold and Maydell, 1978) permits the handling of spatial data in a front-to-back or centre-to-outside order by treating any triangulation as a binary tree with respect to an arbitrary reference (or viewpoint) location.

Figure 1a illustrates the Delaunay triangulation of a rather well-known test data set (Davis, 1973). The numbers represent the order of triangle processing. A viewpoint has been defined , having average X and very large Y coordinates. The processing order (as a binary tree) ensures that triangles closer to the viewpoint are processed before those further away. Figure 1b illustrates the same data set but with the viewpoint located within the

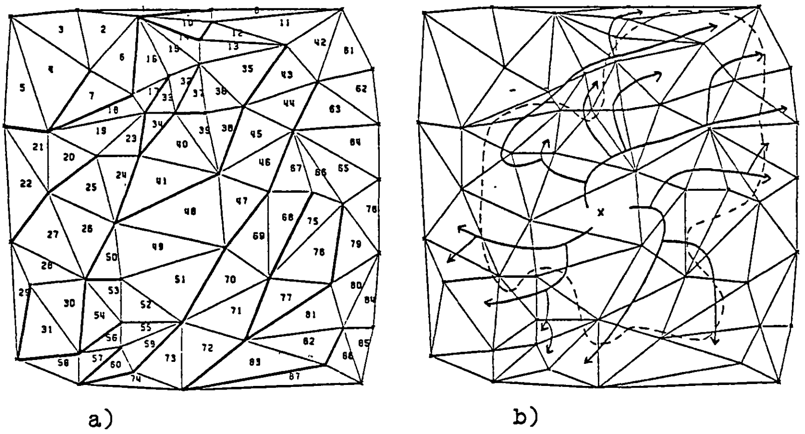


Figure 1.

a) Test data set with triangles, ordered with respect to a viewpoint from the north.

b) The same test data set with a viewpoint interior to the map.

map area. For more details see Gold and Cormack (1986) or Gold and Cormack (in press).

This spatial ordering process permits many useful operations, including: processing a map in contiguous segments; performing line-of-sight, perspective view and hidden line determinations; the extraction of all items likely to be affected in an update of the data base; and appropriate paging of network segments.

APPLICATION TO TERRAIN MODELLING

With the previously described tools at our disposal it is possible to outline an appropriate solution to the generalized perspective view of a topographic surface. The steps are:

1. Insert all data points individually into the database using the Voronoi criterion as described above.

2. Select the viewpoint and field of view.

3. Perform an ordered tree traversal of the triangulated network, starting with data points close to the field of view, and working outwards from there.

4. Reject all triangular facets whose vector normals point "away" from the viewer, as they will have already been hidden by closer portions of the surface.

5. Wherever a closer triangular facet faces the viewer,and its immediately posterior neighbour faces away, a horizon segment has been created. Maintain a radially-ordered linked list of horizon segments. Note that horizon segments will rarely occupy more than a small amount of the active edge of the terrain model at any one time.

6. Where a new forward-facing triangular facet is not behind a current horizon segment it should be drawn. Where it is behind one, it should be determined whether it lies below that horizon or not. If the facet is below the horizon it is not drawn.

7. If the facet is partially above the horizon the visible part of it is drawn, and the appropriate section of the active horizon is deleted. As a consequence of this no facets should occur that are entirely above a currently active horizon segment.

IMPLEMENTATION OF TERRAIN MODEL

In practice, two components are required in addition to the terrain data base software - these are computer graphics routines to permit object rotation and perspective viewing, and an appropriate hidden-line procedure. The transformation routines are available from any computer graphics text. The hidden-line routine that implements steps 5 to 7 may be readily developed. Any hidden-line procedure that maintains a horizon by vector intersection and linked-list maintenance has two properties: it is computationally expensive (hence the importance of eliminating segments as soon as possible, and keeping the active horizon portion short); and it is entirely dependant for its success on the strict preservation of front-to-back ordering of line segments submitted for display. The active horizon consists of those portions of line segments having the largest y coordinate to date (in the screen coordinate system) for any given x coordinate. Any new line segment passed to the routine must be assumed to be further away from the viewer than the previous segments making up the active horizon, thus acting as a clipping window for the new segment. If this is not true, the results make this very obvious - mysterious portions of the final map are blanked out for no apparent reason. Thus this problem is a good example of the strict requirement for ordered spatial processing.

Figure 2 shows the result of submitting a simple test data set to the perspective and hidden-line routines (for the purposes of this presentation lines that would have been hidden are instead drawn lightly). There is, however, one catch. In this example the 28 line segments were ordered manually, taking the viewpoint into consideration - thus Figure 2 shows only that the hidden line routine works. In Figure 3a the interpolation process outlined in Gold (1984)

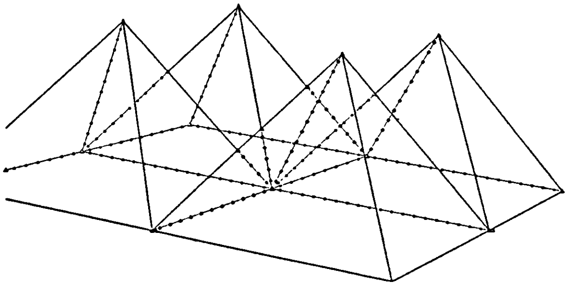


Figure 2.

Four pyramids - line segments viewed in perspective with hidden line removal.

has been used to interpolate elevations at nodes formed by the regular subdivision of the original triangular faces. In this case all line segments were generated by the contouring utilities, and triangles and sub-triangles were ordered away from the viewpoint using the procedures of Gold and Cormack (1986). Since the hidden-line procedure is highly sensitive to line mis-ordering, the spatial ordering procedures previously outlined are clearly effective.

Nevertheless, a problem is evident - perspective depth is not readily discernible: triangular facets do not provide sufficient depth cues. Consequently the surface was re-sampled on a regular grid using the previous interpolation procedure. Note, however, that there is no longer any guarantee that any subsequent surface will match the original data.

This re-sampled, gridded data was then triangulated and displayed as in Figure 2. The result is shown in Figure 3b. Here, however, the desired squares are formed from two triangles. Unlike squares, this gives an unambiguous surface (three points defining a flat plate, not four), but the presence of diagonals reduces the perspective effect. In Figure 3c these diagonals are removed, providing a view in which the perspective cues are satisfactory, even if local surface details are ambiguous. It is salutory to note that the actual information content of Figure 3c is no greater than that of Figure 2. Figure 3d shows a visibility map of the view in Figure 3c.

Thus, unlike some of the better-known discussions of perspective views of grids (e.g. Wright, 1973), it may be

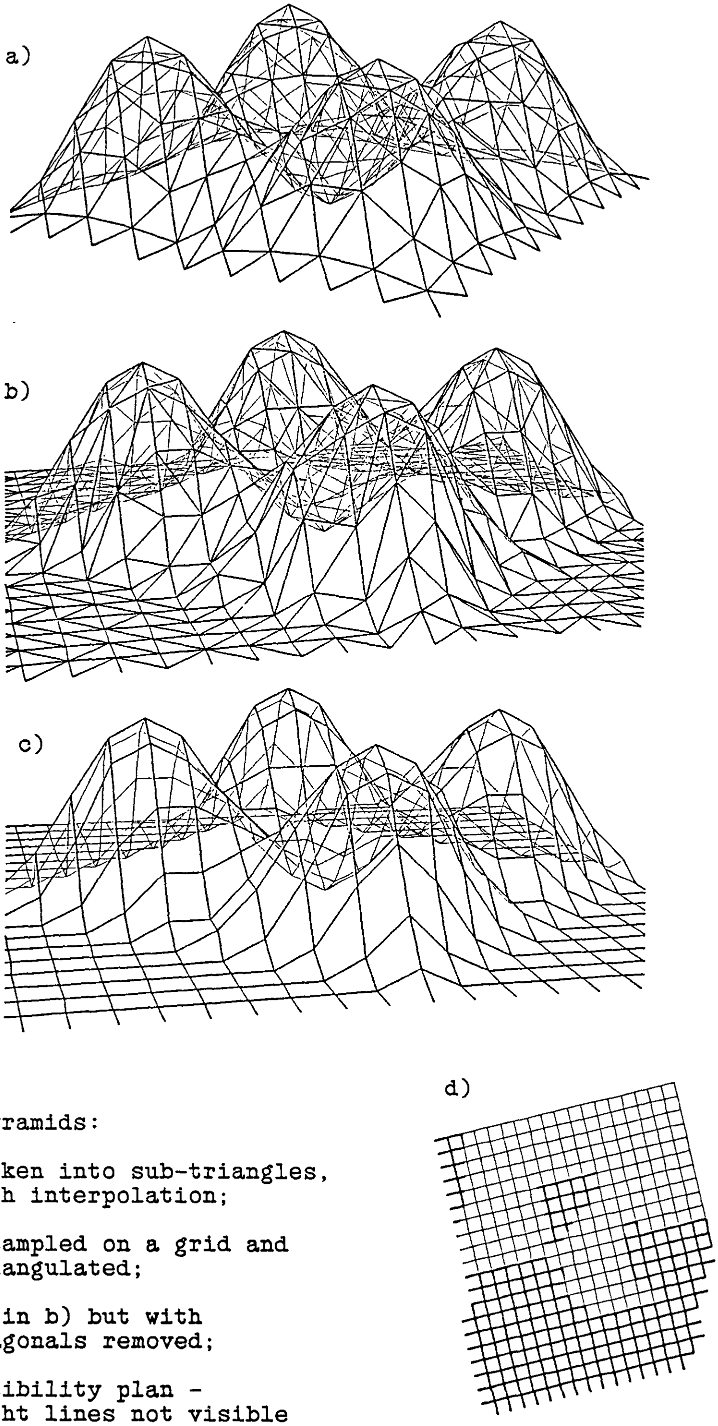


Figure 3.

Four pyramids:

a) broken into sub-triangles, with interpolation;

b) resampled on a grid and triangulated;

c) as in b) but with diagonals removed;

d) visibility plan - light lines not visible from the viewpoint.

convenient to consider grids to be merely special cases of the triangular irregular network (TIN), and the ability to handle TINs in a spatially-consistent fashion permits consistent grid display. The advantages of grids lie in the depth cues. Their disadvantages are the ambiguity of four points forming a surface patch, and all the problems associated with re-sampling data, where that is required.

Ordered triangulations of Davis' test data set were shown in Figure 1. Figure 4a shows a perspective view of this data as a regular TIN. Figure 4b repeats this with triangular subdivision and interpolation as previously described. Figures 4c and 4d show this data re-sampled on a grid and then displayed again as a TIN, with the attendant ordering advantages. In Figure 4c the diagonals are retained, and in Figure 4d they are removed. In Figure 4e the visibility map is shown - this is a by-product of the hidden-line routine. In all cases the hidden-line procedure validates the spatial ordering processes described here.

SUMMARY

On the basis of the outline above, and the previously described spatial ordering procedures, it should be clear that the terrain display process is fairly efficient, as it accommodates a large number of the available spatial relationships. While vector-display perspective views or block diagrams are the most common application, shaded-surface views are equally applicable, as are shadowed-terrain maps, line-of-sight maps and various military applications.

While the generalized procedure is as given above, the special case of perspective block diagrams requires additional comment. Firstly, most block diagrams do not show all horizon lines since the "fishnet" model only shows the outline of the square grid. Since in fact the four corners of a grid are rarely coplanar, horizon (or outline) definition will necessarily be incomplete. To follow the procedure outlined, the squares must be (at least implicitly) subdivided into triangles prior to processing. Only when the horizon does indeed pass through the square does it matter which way the square is subdivided - and only then need the diagonal be actually plotted.

In conclusion, the following points bear repetition. Firstly, surface networks are a tool whose full potential has not yet been realized. Secondly, networks may be manipulated using local O(n) processes with the exception of the global location search function. While theoretical efficiency of the global search can be improved from O(n**1.5), for many applications this is of only marginal benefit. Thirdly, spatial ordering with respect to an arbitrary reference point is of considerable value for both display and data base manipulation. Finally, the example of terrain modelling illustrates how a well-defined set of spatial ordering procedures permits the development of complete, straightforward and efficient algorithms.

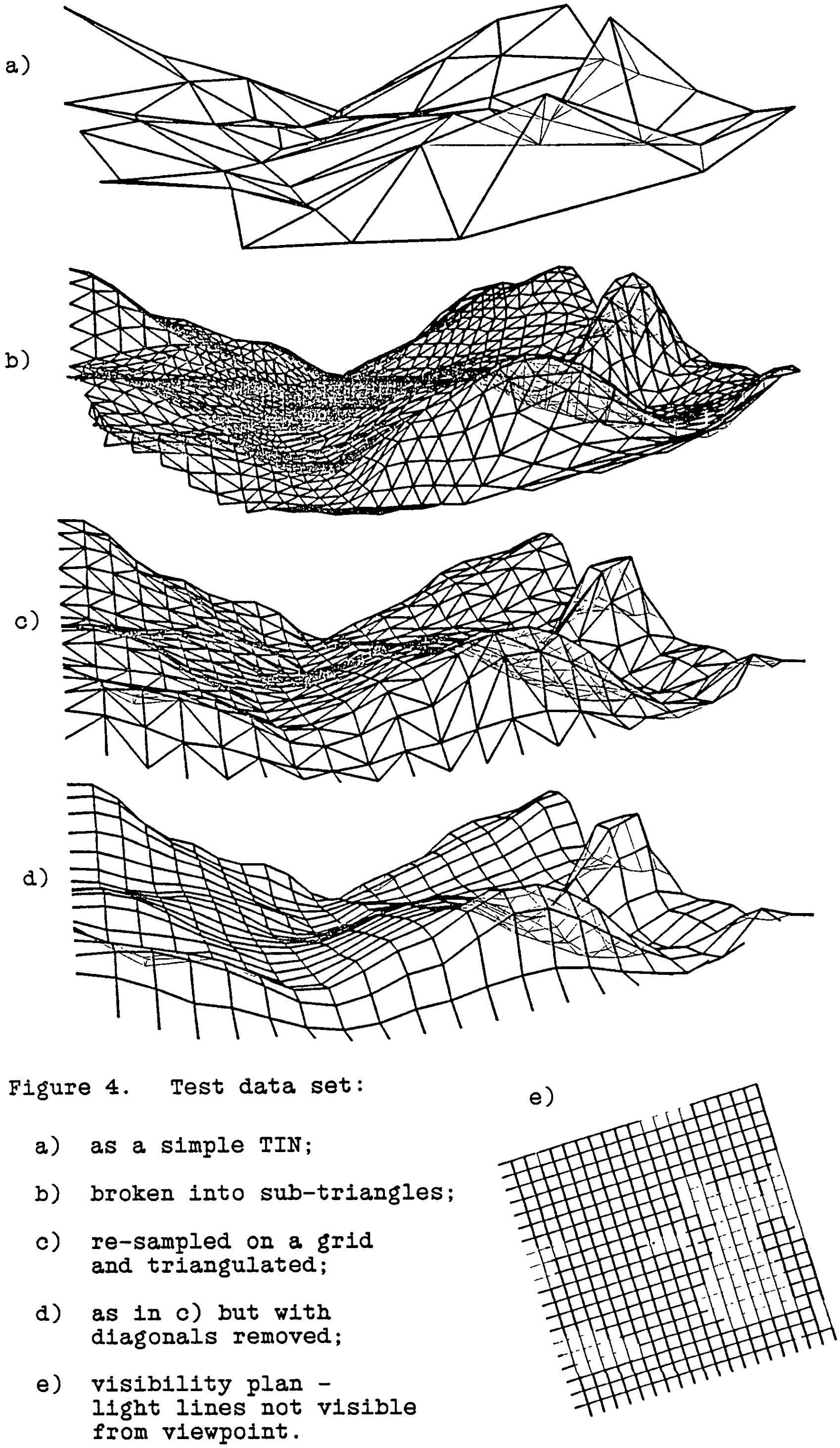


Figure 4. Test data set:

a) as a simple TIN;

b) broken into sub-triangles;

c) re-sampled on a grid and triangulated;

d) as in c) but with diagonals removed;

e) visibility plan - light lines not visible from viewpoint.

REFERENCES

Davis, J.C., 1973, Statistics and Data Analysis in Geology, John Wiley and Sons, New York, p.313.

Faux, I.D., and M.J. Pratt, 1979, Computational Geometry for Design and Manufacturing, Ellis Horwood, Chichester, 331p.

Gold, C.M., 1984, Common Sense Contouring: Cartographica, v. 21 no. 2, pp. 121-129.

Gold, C.M. and S. Cormack, 1986, Spatially Ordered Networks and Topographic Reconstruction: Proceedings, Second International Symposium on Spatial Data Handling, Seattle, July 1986, pp. 74-85.

Gold, C.M. and S. Cormack, (in press), Spatially Ordered Networks and Topographic Reconstruction: International Journal of Geographic Information Systems, v. 1 no. 1, January 1987.

Gold, C.M., T.D. Charters and J. Ramsden, 1977, Automated Contour Mapping using Triangular Element Data Structures and an Interpolant over each Triangular Domain: Computer Graphics, v.11, June 1977, pp. 170-175.

Gold, C.M. and Maydell, U.M., 1978, Triangulation and Spatial Ordering in Computer Cartography: Proceedings, Canadian Cartographic Association Third Annual Meeting, Vancouver, June 1978, pp. 69-81.

Lawson, C.L., 1977, Software for C-1 Surface Interpolation, Jet Propulsion Laboratory Publication 77-30.

O'Rourke, J. 1984, Convex Hulls, Voronoi Diagrams and Terrain Navigation: Proceedings, Spatial Information Technologies for Remote Sensing Today and Tomorrow, Sioux Falls, October 1984, pp.358-361.

Preparata, F.P. and Shamos, M.I., 1985, Computational Geometry an Introduction, Springer Verlag, New York.

Wright, T.J., 1973, A Two Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables: IEEE Transactions on Computers TC22, no. 1, pp. 28-33.

THE USE OF RANGE-TREE SPATIAL INDEXING TO SPEED GIS DATA RETRIEVAL

by
Bruce Blackwell, Chief Scientist
AUTOMETRIC, INCORPORATED
5205 Leesburg Pike,
Suite 1308/Skyline 1
Falls Church, VA 22041

ABSTRACT

Rapid spatial retrieval of data elements is an essential part of an efficient GIS. Many index structures have been used in the past. This paper discusses the use of a new concept, range trees, in these applications. Range trees are well suited to indexing GIS data elements which have finite extents in the 2-D plane and which arbitrarily may be clustered. Range trees are fast and well structured for dynamic disk resident indices. Furthermore, they are readily extensible to multiple dimensions, raising the possibility of volume searches and even extension to attribute space.

INTRODUCTION

Range-trees, hereinafter referred to as R-trees, were first introduced as a spatial indexing strategy for multi-dimensional data by Guttman (1984). Their development was guided by the inadequacy of other indexing methods for handling data elements of finite extent and of arbitrary distribution in a space most common of two dimensions, but more generally of any number of dimensions. Competing index structures include binary trees, cell methods, quad-trees, k-d trees, and K-D-B trees. All of these suffer from one or more severe limitations in geodata applications. Binary trees are based on only one dimension. Even if multiple trees are built to handle more dimensions, retrieval "bands" must be intersected through sequential comparisons to find the desired data elements. All methods require specification of boundaries in advance and are hence inefficient if clustering of data elements occurs, as commonly happens with GIS data sets. Much work has been done on quad-trees and k-d trees, but the application of these structures almost requires that they be implemented in random access memory, since the node sizes are smaller than any common physical disk storage device data block, and are therefore inefficient for disk resident indices. K-D-B trees cannot index data elements of finite spatial extent. R-trees do not suffer from any of these limitations. Furthermore, R-trees are the only indices in current use that are readily extensible to more than two dimensions for special applications.

A full description of the structure, creation, and use of R-trees is contained in the literature (Guttman, 1984). Only a review will be given here, with emphasis on GIS applications. Figure 1 is an example of a portion of a GIS arc-node database. A single arc is highlighted with a bold line, together with a box known as the minimum bounding rectangle (MBR) of the arc. The MBR is a rectangle with sides parallel to the axes of the coordinate space which defines the minimum and maximum spatial extent of the coordinates delineating the data element, in this case an arc. There would also be MBR's associated with nodes and polygons in the database. The MBR of a node is, of course, a degenerate case. MBR's have been used frequently in GIS designs as the basis of spatial research for data elements. It is the MBR's of data entities which are used in the R-tree index. An R-tree is a balanced tree structure wherein each R-Tree node contains a number of entries, and each entry consists of a pointer to a child node and the MBR of the child node. The MBR of a node is the least rectangle containing the MBR's of all its children. Two levels in the R-tree have special significance. There is a single node at the beginning of the tree called the root. At the end level of the tree, the nodes are called leaves and the child pointers are to database entries themselves rather than to lower level nodes in the tree. Recursive algorithms for initially populating, updating, and searching the R-tree have been well defined (Guttman, 1984). A small R-tree example is shown in Figure 2.

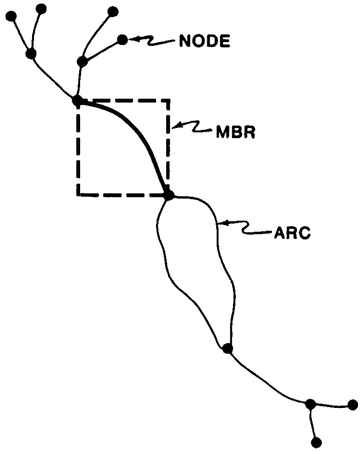


Figure 1. R-tree Example

Two aspects of R-trees are particularly important in applications: node size and node-splitting. Two parameters determine the number of child entries in each node; m, the smallest number of entries allowed, and n, the largest number allowed. No node, except the root, can have fewer than m entries. The parameter n is critical to input/output efficiency for disk-resident R-trees and should be chosen so that the physical disk block size B is given by:

$B = (S_{MBR} + S_{PTR})\ n + O$, where:

S_{MBR}, S_{PTR} are the storage required for the child MBR and pointer entries;

O is the overhead in each node, flags, etc.; and

B typically ranges from 256 to 2048 bytes for different direct access storage devices.

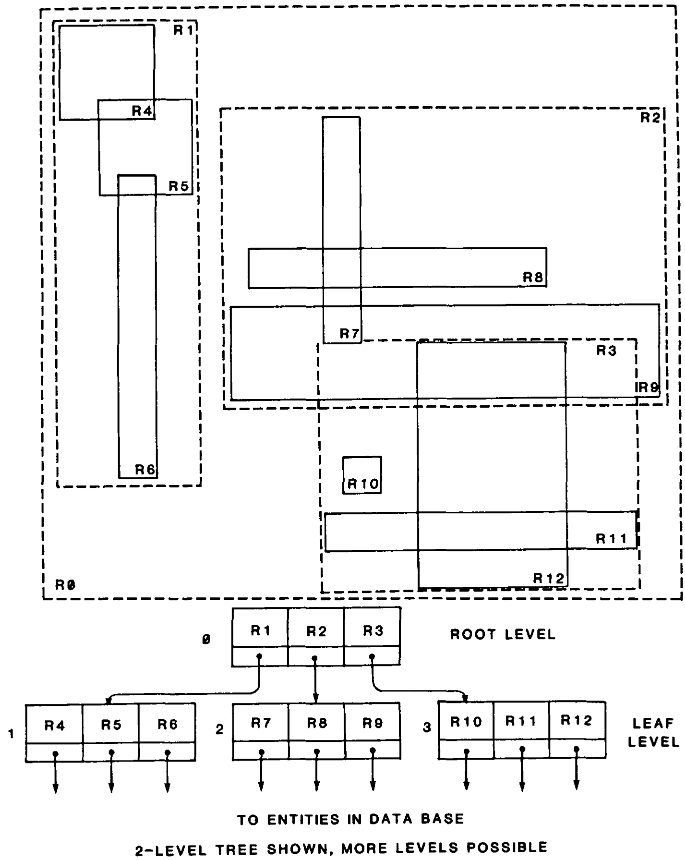


Figure 2. R-tree Spatial Search Index

When a new entry is to be made in the tree, the algorithm (Guttman, 1984) picks the leaf node for insertion which would have to be enlarged least to accommodate the new entry. If this leaf already contains n entries, it must be split. Node-splitting is critical to future search performance of the tree. Exhaustive, quadratic cost, and linear cost algorithms were explored by Guttman (1984). In my laboratory, I have employed a modified form of Guttman's linear cost algorithm with excellent results. Recently, it has been suggested (Roussopoulos and Leifker, 1985) that for relatively static spatial databases, a special packing algorithm be employed for initial population of the R-tree, since Guttman's recursive insertion can lead to inefficiencies in terms of total coverage area and coverage overlaps. While true, it has been my experience that in a GIS production environment the R-tree is built from the bottom for each geounit during data extraction, meets interactive search response time requirements, and the commonality of software between initial population and dynamic updating is a great advantage. There is nothing to prohibit later restructuring of the R-tree with optimized packing algorithms if desired, once a condition of data status is achieved.

EXPERIENCE WITH R-TREES IN A GIS LABORATORY

Application

Autometric, Inc. is working with a GIS in its laboratory which is a fully topologically integrated arc-node data structure. Cartographic information, that is, features, are maintained in a separate set of records which link component topological node, arcs, and polygons. Stored with the features, or in a companion relational database, are feature attributes. The topological arcs and nodes carry the spatial component of the information. For each geounit, then, there are two separate file aggregates which are networked together, the topological/spatial and the feature files. Each entity in each file has an MBR associated and stored with it. The MBR of a feature is derived by combining the MBR's of its component topological parts.

R-trees are built for both the topological and feature files as each element is entered in the database interactively at the workstation. The R-tree serve several purposes:

Display Windowing. The user is able to select, via cursor or keyboard entry, a new display window. The geographic coordinates of the new window are used as input to an R-tree search to determine the list of entities to be displayed. If a symbolized display is desired, the search proceeds through the feature R-tree, and graphics lists are derived subsequently by following links to the topology. If a topological network display is desired, the search proceeds through the topological R-trees.

Cursor Location in the Topology. Formation of topology is done "on-the-fly" while digitizing or editing from photo or cartographic source. The operator can place the cursor anywhere in the geounit area, create a node, snap to an existing node or arc, and digitize arcs. The topological R-tree is used to rapidly obtain such information as the containing polygon, the nearest node, or the nearest arc. Upon completion of digitizing an arc, the arc MBR is built up from its component spatial coordinate list and is passed through the R-tree to retrieve any other arcs in the neighborhood. These arcsare tested for intersections with the candidate arc. If the arc is acceptable, it is entered in the database and in the R-tree. If the arc divides a polygon, two new polygons are formed and the old is deleted, both in the database and in the R-tree. The R-tree, therefore, serves as an adjunct to database navigation and edit.

Entity "Pick" Functions. Cartographic features are assigned to component topological entities by picking nodes, arcs, or polygons by placing the cursor on or near them. The R-tree is used to find rapidly one or more candidates in response to the "pick" request. The candidate(s) is then checked on the basis of a spatial tolerance before being assigned to the feature.

Implementation

In the Autometric GIS laboratory, R-trees have been implemented, along with other GIS functionalities, on a VAX 11/750 computer. The workstation consists of an APPS-IV analytical plotter with graphic superpositioning, an ALTEK digitizing table, a LEXIDATA color graphics terminal, and an A/N CRT and keyboard. The R-trees are disk-resident with a physical record size of 256 bytes so that each

R-tree node has a maximum of 12 entries. A typical data set size for a GIS geounit is 30,000 to 50,000 topological entities and 8,000 to 12,000 cartographic entities.

Performance

With the data set size and R-tree organization described in the above paragraph, virtual windowing and entity pick functions can be performed with response times adequate to support interactive operations. The R-tree search time is well described by the following relation:

$$T_R = A\ m \log n + B$$

where

T_R = retrieval time, wall clock

m = the number of retrieved entities

n = the total number of entities in the R-tree index

A = a constant

B = b constant

In our application, A is about 2.5×10^{-4} seconds and B is about 4 seconds. Note that the search time is linear in number of retrieved entities but logarithmic in the size of the database. The slow search time growth with tree size is a main advantage of tree structures.

Future Possibilities

Most of the nodes in an R-tree are at the leaf level, but most of the node traversals during search, intersection, and deletion take place at levels in the tree above leaf level. An obvious R-tree usage speed improvement could be obtained by holding all R-tree nodes above leaf level in random access memory. The number of node entities above leaf level is approximately equal to n/m (m-1) where n is the total number of database entries and m is the average node fill rate. For n = 50,000 and m = 10, the number of nodes above leaf level in 555, and these can be held in 140 k bytes of memory. Preliminary work reveals that a performance improvement of between 6 and 10 to 1 is possible with this mechanism.

R-trees are in no way restricted to two-dimensional spatial indexing. They generalize readily to a space of an arbitrary number of dimensions. An obvious extension is to include the elevation of features in a 3-D minimum bounding rectangular solid (MBRS) and use the MBRS's to build an R-tree index. Such an index might be useful for the use of a GIS for the production of special products such as air navigation hazard guides. Less obvious is the possibility of treating attributes, properly hierarchically coded, as a "dimension" of an abstract space, and including the attribute dimension in the R-tree index. Such an implementation would permit rapid simultaneous spatial and cartographic layer retrievals from the database. For example, a query to obtain and display all hydrologic features in a horizontal zone could be quickly processed.

REFERENCES

Guttman, A. 1984, R-Trees: A Dynamic Index Structure for Spatial Searching: Proc. of ACM SIGMOD Conference on Management of Data, Boston, June 1984.

Roussopoulos, N. and D. Leifker 1985, Direct Spatial Search on Pictorial Databases Using Packed R-trees: ACM Transactions on Database Systems, Vol.5, 1985, pp. 17-31.

FORMATTING GEOGRAPHIC DATA TO ENHANCE MANIPULABILITY

Gail Langran
Mapping, Charting, and Geodesy Division
Naval Ocean Research and Development Activity
NSTL, Mississippi 39529

Barbara Pfeil Buttenfield
Department of Geography
University of Wisconsin
Madison, Wisconsin 53705

ABSTRACT

Geographic data tends to be exploited extensively and imaginatively once it becomes available. When standard data sets serve as input to applications software, however, the data must often be filtered or restructured. Given this likelihood, special attention should be paid to any distributed data set's manipulability. This paper discusses ways to organize sequential data sets to facilitate three major filtering tasks: windowing, categorical feature selection or aggregation, and resolution reduction Examples are drawn from current format standards.

INTRODUCTION

Users of mapping and geographic information system software can select their systems' input from a small but steadily growing assortment of digital geographic data sets. Since data is a far scarcer commodity than software, a seller's market has resulted. Not surprisingly, data vendors have chosen to distribute their data in standardized forms so resources may be directed to capturing data, rather than diverted to tailoring customized versions of data sets.

Since most applications will manipulate the standardized data format to meet user-specific needs, it follows that manipulability is a desirable data characteristic. The problem, then, is twofold: first, how does one predict which manipulations will be performed on a given data set? And once these are predicted, how can the data set be designed to facilitate the manipulations? The next section illustrates a method of extrapolating potential data set manipulations, followed by a discussion of ways to facilitate filtering of sequentially ordered data using current format standards as examples. The final section summarizes some of the points made here and suggests further work.

FORECASTING DATA MANIPULATIONS

This exercise takes four data set types--shoreline vectors, cartographic features, navigational chart data, and a digital elevation model--and envisions a set of applications for each. Once an application is forecast, its component operations are extrapolated.

World Shoreline Vectors

Shoreline vectors are the most venerable of all cartographic data sets. While commonly used in the past to sketch background maps, their current applications have broadened considerably (Table 1).

Table 1. Applications and (manipulations) of world shoreline vectors.

- Route planning: plot a route by air or sea that avoids a given country or region lying between its two endpoints.

  (create a land mask, apply topological constraints)

- Distance computations: compute the distance from nearest landfall to current position at sea; compute distance from port to current position at sea.

  (compute point-to-point or point-to-line arc distance)

- Merge data: add features; add bathymetric data.

  (transform coordinate system, translate feature codes)

- Repartitioning and windowing: group segments by oceans instead of by continents; extract an area of given dimensions around a given point; extract an area whose corners are given; extract an area that will fit on a given display device at a given resolution.

  (search, extract, clip)

- Restructuring: extract and use spaghetti data only; add adjacency information.

- Scale change: enlarge or reduce.

  (generalize or enhance lines; generalize small island and lake groupings)

- Display: create a map image.

  (label; symbolize; draw outlines only, color fill, merge features from a feature file)

Feature/Attribute Files

For our purposes, features and attributes are defined to include transportation and communication networks, political boundaries, drainage, hydrographic data, vegetation, and other mappable point, line, or areal data in polygon form. Possible uses for such data are listed in Table 2.

Electronic Navigation Charts

A growing family of electronic navigation charts share several properties: many functions occur in real time, some data is received in real time from sensors, and a default mode leaves few cartographic choices to the user. Table 3 extrapolates data manipulations for a shipboard electronic chart. Automotive applications are also possible.

Table 2. Applications and (manipulations) of feature data.

- Spatial comparisons: determine the adjacency, overlap, or distance between features.

  (compute point-to-point or point-to-line distance)

- Feature selection or aggregation: group all drainage features into one feature type rather than discriminating between rivers, streams, and canals; group deciduous, conifer, and mixed forest type into a single forest type; group individual hazards to navigation as "hazards" or different types of obstacles to aviation as "obstacles."

  (search for features; match features to segments; extract)

- Attribute selection, aggregation, or ordinal grouping: group all lighted harbor buoys into one category regardless of light color or strobe frequency; group all vertical obstructions over a given height into the category "hazard to aviation;" rank vertical obstructions into height categories; rank towns by population.

  (search for attributes; delete or assign new codes)

- Repartioning and windowing: see Table 1.
- Restructuring: see Table 1.
- Scale change: see Table 1; also, reclassify area features as point or line features.
- Display: see Table 1.

Table 3. Applications and (manipulations) of electronic navigation charts.

- Real-time computations: use current position, speed, bearing, chart features, and sensor input (e.g., radar, sonar).
- Feature selection and aggregation: see Table 2.
- Attribute selection and aggregation: see Table 2.
- Repartitioning and real-time windowing: see Table 1.
- Restructuring: see Table 1.
- Real-time scale change: see Table 1.
- Display: see Table 1; also, real-time animation and update.

Gridded Data

In the context of data manipulations, elevation, bathymetric, and other types of gridded data differ considerably from the previous three examples. The gridded format persists precisely because it is easily manipulable; far more space-efficient structures have been overshadowed by the programmability of the gridded format. Gridded data sets are therefore included in this discussion (Table 4) for contrast.

Table 4. Applications of gridded elevation data.

- Windowing: compute location based on pole spacing, extract.
- Scale change: eliminate elevations (i.e., reduce grid resolution); interpolate new points (raise grid resolution).
- Display: compute contours, apply hillshading, layer tints, or other graphic effects.

Summary

Despite differences in application and content, a common thread runs throughout the data sets listed above: each may be filtered via windowing, categorical selection, and reduction of scale or resolution which, from this point on, will be referred to simply as reduction. Other possible manipulations include repartitioning and restructuring. Only the filtering process will be examined here; however, the methods and philosophy apply to all forms of data manipulation.

Given the importance of filtering to geographic data sets, the question arises: are current and planned data sets organized in a manner that is maximally conducive to such filtering? The next section uses some of today's standard formats to explore this topic.

MANIPULABILITY OF TODAY'S FORMAT STANDARDS

The discussion that follows suggests ways to facilitate the three major filtering operations and compares how each is addressed by today's standards (Table 5). A fourth performance factor, programmability, is considered last.

Table 5. Acronyms of referenced formats.

Acronym	Full name and (sponsor)
CEDD	Committee on Exchange of Digital Data (International Hydrographic Organization)
DEM	Digital Elevation Matrix (USGS)
DLG-O	Digital Line Graphic - Optional (USGS)
FGEF	Federal Geographic Exchange Format (Federal Interagency Coordinating Committee on Digital Cartography)
GDIL	Geographic Data Interchange Language (Jet Propulsion Laboratory)
GIRAS	Geographic Information Retrieval and Analysis System (USGS)
MCDIF	Map and Chart Data Interchange Format (Ontario Ministry of Natural Resources)
NCDCDS	National Committee on Digital Cartographic Data Standards (sponsored by the same)
SDDEF	Standard Digital Data Exchange Format (NOS)
SLF	Standard Linear Format (DMA)

Before continuing, however, it must be emphasized that some of the formats referenced in this section (CEDD, FGEF, SDDEF, SLF) were designed as vehicles for the exchange of mapping data among or within map production agencies. While these formats were never intended to be manipulable, it is yet instructive to examine them. Other formats (DLG, GIRAS, DEM, WDB II) were designed for more general use. A final class of format standards will not be discussed here. Such formats are essentially virtual envelopes into which data is sealed for dissemination. The envelopes describe the characteristics of the data contained within via coordinate transformation parameters and format statements that facilitate data loading. While extremely useful, the virtual envelopes (GDIL, NCDCDS, and MCDIF) are not relevent to this discussion and will be excluded.

Repartitioning and Windowing

Most commonly, windowing is a straightforward process of subtraction: find and collect only the segments that overlap a given area, then clip from those segments the pieces that are outside the window. Processing is proportional to the number of line segments being searched. Thus, timing problems arise as file size grows. The amount of data packed within a given file unit is of obvious importance in windowing or partitioning efficiency. DLG and GIRAS files reflect this constraint. The 1:100,000 DLG files are subdivided into 15' or 7.5' cells depending on data density. GIRAS files are subdivided into sections not exceeding 32,000 coordinate pairs or 2500 arcs. While the subdivisions were adopted due to memory constraints, their effect is improved windowing performance.

While a slight reduction in file unit size improves the efficiency of subtractive windowing, systematically subdividing the file into small rectangular cells allows users to adopt an additive method of windowing or repartitioning. For optimum results, cell size should match that of the smallest area to be windowed. To window, all cells that comprise the desired window (or partition) are assembled and adjoined, thus avoiding arduous segment searches and clipping. This method brings two space-saving bonusus: if computer memory is constrained data can be loaded in small pieces, and if coordinates are stored relative to cell origins file size is reduced.

To structure a file into cells, segments are clipped and nodes are formed at cell edges. Information concerning cell dimensions is recorded in the volume header, and short cell headers are constructed to provide cell origins (cell coverages are computed using the volume header information). To expedite the search for desired cells, a cell code can be computed and placed in the header to allow spatial hashing based on latitude and longitude (Connor and Langran, 1987). Alternatively, users have the option of aggregating to larger cells or to a quadtree cell representation (Jones and Abraham, 1986).

Categorical Feature Selection

Considerable machination may be needed to extract from a standard file the particular feature and attribute classes desired for a given application. Conceptually identifying the necessary features can, in itself, be a problem, since three feature coding standards exist in U.S. mapping agencies alone (DMA, 1985; NOS, 1985; and USGS, 1985) and a fourth is being recommended by NCDCDS (1986).

The NCDCDS recommends a hierarchical classification scheme for features and attributes that casts major feature types as nouns that are modified by attribute "adjectives". Both USGS and DMA's coding schemes reflect this sentiment to some degree. The USGS' 3-byte major code is a broad category--e.g., water bodies, political boundaries, rivers and streams--while its 4-byte minor code is descriptive: single-line perennial stream of length 50-60 km, perennial lake or pond, etc. DMA's 5-byte coding scheme describes category and subcategory in the first and second characters, respectively. The broad category represented by the first character (e.g., hydrography) leads to a more specific subcategory (e.g., ports and harbors," "dangers and hazards," "bottom information"). The final three characters are assigned sequentially to features in alphabetical order.

Once the user transcends terminology differences, he must write software to extract from the sequential file the feature subset he needs. The general procedure is:

1. Encounter a feature.
2. Determine the feature's processing needs.
3. If processing is needed, process the feature.

Step 2 stands out as an area where data adaptation could be helpful. Tabular and hierarchical methods of determining processing needs are possible. The tabular method constructs a look-up table containing the codes and processing needs of features to be included. The algorithm is:

1. Search for the feature code in the look-up table.
2. If found, reference and perform the required processing.
3. Get the next feature.

This procedure would be facilitated if feature codes were available in digital look-up tables, which could be edited as necessary by the programmer. Lacking digitized tables, programmers nationwide must do a great deal of duplicative typing.

A cascading method is possible for hierarchical coding schemes. A user may wish to extract from DMA's Hydrography category all port and harbor information, to exclude all bottom type information, and to aggregate all hazards into one "Dangerous" class. The algorithm is:

1. Read the first digit of the feature code.
2. For a hydrographic code, read the second digit.
3. For a port and harbor code, continue reading digits to obtain the rest of the data detail.
   For a hazard code, call the feature "Dangerous" and load it into the data base.
4. Get the next feature.

This method is particularly useful when elimination or regrouping occurs at the categorical level. Without hierarchically assigned feature codes, however, it cannot be used.

Reduction

The previous subsection described categorical, or qualitative, filtering. Reduction implies that features are eliminated based on spatial and quantitative factors: the feature is not important enough to crowd the map at the intended display scale.

Two major operations occur in reduction: points are eliminated from lines and areal boundaries, and features are eliminated based on space available and relative importance. We could find no evidence that any standard format has incorporated ways to facilitate either generalization operation. Since none are in use, this section discusses the feasibility of several data adaptations.

Line segments can be stored hierarchically, although the referencing system would add to data set size. Ideally, hierarchies would be based on geographic features so critical points are preserved in node values. To date, only rudimentary methods of recognizing linear feature types exist (Buttenfield, 1987). Assuming a tolerance-based line generalization strategy similar to the Douglas algorithm (Douglas and Peucker, 1973), tolerance values could be stored in feature look-up tables to avoid the poor results of generalizing all features uniformly (Buttenfield, 1986). Where positional integrity is required, flags could be embedded in segments to denote points that must not be altered due to navigational or other importance.

How to package sequential data to facilitate the second type of operation is problematic. WDB II stores ranks with island and lake groupings, so smaller islands can be deselected as scale is reduced to avoid coalescence. A more flexible alternative might be to store areas or population values with such features so users can determine their own rankings.

Processing efficiency

Processing efficiency can be defined as a rational balance in use of space and time. Programmability, a third factor, is gaining in importance as human resources grow more expensive relative to computer resources.

The physical and logical arrangement of data upon media has a major impact on processing efficiency. A good example can be drawn from logical and physical blocking of tapes. Table 6 shows the impact of block size on a tape's storage capacity. Since blocks must be physically separated on tape by interblock gaps, large blocks with few separa- tions are far more space-efficient. Large blocks are also more time-efficient, since it reduces the number of times the input program must access the tape.

Table 6. The impact of block size on storage capacity. Values are computed for a typical 2400-ft tape using a 0.75-inch interblock gap.

block size	#blocks	Tape capacity at 1600 bpi
8000 bytes	5008	40 MB
5120 bytes	7291	37 MB
1980 bytes	14490	28 MB
1280 bytes	18580	23 MB

The logical organization of records within blocks is a space and programmability issue. Small records are generally used, since these require less padding with spaces and are easily viewed at a terminal. Programmability becomes a further problem when logical records cross the boundaries of physical blocks, as is the case with SLF and CEDD (Table 7).

The use of fixed or free format trades processing speed against flexibility. Since fixed formats are essentially read by template and free formats must be parsed, speed differences can be considerable. Fixed formats include SLF, DLG, SDDEF, CEDD, and WDB II. FGEF is of free format; users define a set of delimiters in the header to separate records, fields, and subfields. An interesting hybrid is proposed by NCDCDS, which would have the computer parse for format statements, which are then used to read N bytes of data in fixed format. GDIL suggests placing these format descriptions in a file header.

Space does not permit a full discussion of these data processing issues. However, further examples can be drawn from coordinate treatment, binary vs. ASCII storage, and media type. Often, the designer must choose between maximizing space, time, or programming efficiency. Since applications users may be constrained in all three areas, the right choice will require a careful deliberation.

Table 7. Size of logical records and physical blocks specified for standard data formats.

Format	Record size	Block size
CEDD	(1)	1980
DEM/DTED	(2)	-
DLG-O	80	multiple of 80
FGEF	80	1280
GIRAS	80	multiple of 80
SDDEF	80	5120
SLF	multiple of 1980	1980
WDB II	80	8000

(1) CEDD specifies four record types: the header (565 bytes), features (188 bytes), segments (42 bytes), and text (1972 bytes).

(2) DEM/DTED files have three record types: the header (864 bytes), the data (144 + (rows * columns * I6)), and data quality statistics (60 bytes).

SUMMARY

A broad range of topics have been discussed. Our original questions concerned how to adapt data so it is more amenable to reformatting by applications software. The paper's method is largely exploratory and expository, since few attempts have yet been made to design manipulability into sequential data sets. Since a number of sequential exchange formats are currently in formative states, however, such ideas could be incorporated with relative ease. If the future of geographic information processing includes data exchange with those outside the mapping profession, a wider range of applications, and a great deal of preprocessing, should be expected and planned for.

REFERENCES

Billingsley, Fred and Strome, W. Murray (1986). "Standardization of Remote Sensing and GIS Data Transfer." Paper presented at the ISPRS Convention, Baltimore, Maryland, May.

Buttenfield, Barbara Pfeil (1987). "Automatic Identification of Cartographic Lines." The American Cartographer (in press). January.

Buttenfield, Barbara Pfeil (1986). "Digital Definitions of Scale-Dependent Line Structure." Proceddings of Auto-Carto London, September. Vol. 1, p. 497-506.

Connor, Maura and Langran, Gail (1987). "Spatial Hashing to Facilite File Windowing." Naval Ocean Research and Development Activity, NSTL, MS. (in press).

Defense Mapping Agency (1985). "Standard Linear Format." Washington D.C.

Defense Mapping Agency (1985). "Feature Attribute Coding Standard." Washington, D.C., July.

Douglas, D. H. and Poiker (formerly Peucker), T. K. (1973). "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature." Canadian Cartographer Vol 10(2), p. 110-122.

IDON Corporation (1986). "MACDIF: Concept Definition." Submitted to Canadian Hydrographic Service, Document number ACN 3-R&D-068-CD, July.

IDON Corporation (1986). "MACDIF: Structure and Coding (draft)." Submitted to Canadian Hydrographic Service, Document number ACN 3-R&D-068-SC, November.

International Hydrographic Organization (1986). "Format for the Exchange of Digital Hydrographic Data." By the Committee on the Exchange of Digital Data, November.

Jet Propulsion Laboratory (1986). "General Data Interchange Language." Pasadena, California, May.

Jones, Christopher B. and Abraham, Ian M. (1986). "Design Considerations for a Scale-Independent Cartographic Database." Proceedings of the Second International Symposium on Spatial Data Handling, Seattle, July.

Langran, Gail; Connor, Maura; and Clark, R. Kent (1986). "Recommendations on DMA's Standard Linear Format." Naval Ocean Research and Development Activity, NSTL, MS, NR 146, July.

National Ocean Survey (1985). "Charted Features Data Base Categories: Feature Category Keys." Rockville, Maryland.

National Ocean Survey (1985). "Standard Digital Data Exchange Format." Rockville, Maryland, March.

U.S. Geological Survey (1983). "Digital Line Graphs from 1:2,000,000-Scale Maps." Reston, Virginia, Circular 895-D.

U.S. Geological Survey (1983). "Digital Elevation Models." Reston, Virginia, Circular 895-B.

U.S. Geological Survey (1983). "Land Use and Land Cover Digital Data." Reston, Virginia, Circular 895-E.

U.S. Geological Survey (1985). "Digital Line Graphs from 1:100,000-Scale Maps." Reston, Virginia, Data Users Guide 2.

HOW TO SURVIVE ON A SINGLE DETAILED DATABASE

M. Kate Beard
Department of Landscape Architecture, 25 Ag Hall
University of Wisconsin-Madison, Madison, WI 53706

ABSTRACT

This paper reports on an approach to a data base that assumes only a single detailed coverage is required in place of separately derived coverages at more generalized scales. Less detailed or smaller scale representations are produced by manipulation of the detailed coverage. The suggested approach requires scale changing and generalization tools to handle large reductions of detail. This approach is demonstrated using nautical chart data ranging from 1:10,000 to 1:40,000 scale. Some initial results are tested by comparison to an independently generalized product.

INTRODUCTION

It would be ideal for small scale maps to be derived from large or medium scale maps (Watson 1970). In such a case only a single survey would be required. In a digital environment, it should be less time consuming, less costly and potentially less error prone to convert, store, and maintain a single large scale coverage and derive all smaller scale coverages. In practice this approach has rarely been followed. The conventional order of national mapping priorities is to achieve complete coverage at small or medium scales first since these can be completed most rapidly. The USGS digital mapping effort epitomizes this priority scheme. Although their original intent was to start with the 1:24,000 series, the actual order of production was the conversion of the 1:2,000,000 scale series followed by conversion of the 1:000,000 scale sheets. Their future plans include digitizing the 1:24,000 scale series which will assure that a digital version of each scale will exist. Given the capabilities of digital mapping at the time this effort was started their order of production is perhaps understandable. Early systems were not well equipped to handle the bulk of large scale mapping nor did the Survey probably wish to risk a large investment on still experimental technology.

There are, however, sufficient disadvantages in conversion and maintenance costs, and map series inconsistencies that this approach should not be perpetuated. An alternative approach is to convert, and maintain a single detailed coverage. Once this coverage is complete it should be sufficient. All smaller scale coverages could then be derived from this source. Implementation of this approach, however, is not a simple matter. Detailed information is time consuming and expensive to collect, and once the data is collected the conversion to any smaller scale is not a trivial matter. A single coverage approach requires considerations far beyond line generalization algorithms and factor of 2 scale reductions. This paper begins with a discussion of the advantages and disadvantages of this approach then proposes a data base design and collection of automated generalization tools that would be required to carry it out.

ADVANTAGES OF A SINGLE COVERAGE APPROACH

Reliance on a single detailed data base has several advantages. It eliminates the time and cost of converting several smaller scale versions of the same coverage. It simplifies and reduces the cost of maintenance and updates since an update need only be applied to one coverage. Inconsistencies due to time lags in updating several different scale coverages are removed. Also inconsistencies due to application of different source material to different scale versions could be eliminated. If a range of scales are to be converted and stored the question will naturally arise as to how many versions are stored and which ones. At one extreme there could be as many scales as there are users or applications. The storage situation could easily get out of hand. Adoption of a single coverage avoids this situation.

DISADVANTAGES OF A SINGLE COVERAGE APPROACH

There are difficulties in implementing a single coverage approach a number of which are identified in the following quotations.

> Of fundamental importance for generalization processing is the issue of whether the spatial data are digitized and stored once or several times for different scales. Theoretically, this represents a trade-off between processing and storage requirements. In practice a single data base solution will not be feasible for several reasons; time delay for real time applications, lack of generalization procedures, scale-related variability of objects to be stored. (Brassel 1985 p.22)
>
> Small scale retrieval of lines stored at a single large scale involves incurring very large overheads, both in the quantity of data accessed and in processing by the generalization algorithm. (Jones and Abraham 1986 p.388)
>
> There is no overall program of large scale digital base mapping and as yet no suitable base is readily available. To be suitable such a base would have to exist at a number of scales since it would be too elaborate and expensive for users to derive a specific scale by generalization from a largest common denominator scale. (Tomlinson 1986 p.10)

The main disadvantages raised in the above comments, concern the size of files, excessive processing times, and inadequate treatment of scale-related variability within a file. These objections may be valid given current thinking and the status of automated generalization procedures in current production systems. A solution exists, however, in a rethinking of data storage arrangements and improvement of scale changing and generalization algorithms.

A SOLUTION TO THE PROBLEM

One part of the solution would require changes in the institutional responsibility, creation, and format of a detailed database. The second part requires software development. A first task, however, is to define the term detailed coverage. This definition is then followed by discussion of data base changes and software improvements.

A Detailed Coverage

A coverage is a layer of one theme such as soils, hydrography, or land use. A coverage is based on a classification scheme and constructed by identifiying the spatial location of a class and delineating it from neighboring classes (Chrisman 1982). A detailed coverage implies detail in both the geometric and attribute components. Spatial detail (in a vector representation) will depend both on the number of points allowed to approximate boundary curves and the number of significant digits used to store the coordinates. For the attribute information greater detail is associated with a larger number of classes and finer discrimination between classes. Such a coverage should not be confused with a general purpose base map. The traditional base map attempts to anticipate the needs of several users by including all manner of information on a single map.

A detailed coverage cannot be tied to a specific scale, but will correspond generally to a large scale mapping. Detailed coverages may range from 1:200 to 1:50,000 scale depending on the nature of the resource or landscape and current knowledge of it. The level of detail or scale will depend basically on the size of the objects to be represented. A 1:50,000 scale map of bedrock in the Midwest might be considered detailed because of the size and homogeneity of the rock bodies. A detailed coverage of Rocky Mountain bedrock geology might require a larger mapping scale to depict the greater geological complexity. Similarly a detailed coverage of urban land use might mean a map scale of 1:500 while a detailed rural land use coverage might mean a 1:4,800 map scale.

Data base design changes

Distributed collection and maintenance of data. The first change is an institutional change in which responsibility for mapping would be delegated to responsible agencies at state, regional or local levels. Agencies at these levels are the prime users of detailed data and therefore should have a vested interest in its collection and maintenance. Detailed information is time consuming and costly to collect for large areas, but, if responsibilty for collection was dispersed the burden would be less for each contributor. The detailed data base could be built up incrementally by first focusing on selected areas such as urban, rapidly developing, or high environmental risk areas, then filling in the gaps as time and budgets allowed. State, regional, and local governments would be in a more logical position than the federal government to set such priorities. The data base could also be built from existing pieces such as detailed information collected for specific projects. As an example, detailed geologic information collected for the siting of a nuclear waste repository could be spliced into a more general coverage. Scale uniformity should not be a constraint if topological consistency is maintained. A larger scale or more detailed inset would simply mean a greater coordinate density for that area.

Substitution of detailed coverages for base maps A second change is to replace the general purpose base map with individual coverages of detailed information. A coverage of information, while detailed, should not create as large a storage and processing overhead as a general purpose base map with information on every conceivable object in the landscape. Maintenance of coverages also eliminates the problem of scale-related variation among different features since a coverage will include only one feature type. Hydrography treated independently should be simpler to generalize than a base map which includes roads, buildings, vegetation and contours in addition to hydrography.

Unfortunately the notion of a base map has become entrenched as a necessary foundation for all mapping activity (Bauer 1983). The base map provides a manual merger of several layers of information and has endured since the merger of digital layers has not been an efficient automated capability until recently. With the development of more sophisticated systems, and the availability of a geodetic reference framework and identifiable control points for each coverage, a base map may no longer be necessary (Chrisman and Niemann 1985).

Storage of data in practical analytical units There is no rule which requires a detailed coverage to exist as one large file. An alternative is to store data in manageable units based on obvious political or natural boundaries. The units could be counties, townships, watersheds, etc. depending on the information, its scale of variation, the level of detail currently available, or a legislative mandate. As the level of detail increases, a smaller areal unit could be used. If the size of the storage unit is well matched to the level of detail, processing times should not be unreasonable. Experience with the Dane County Land Records Project has shown that detailed (1:15,840 scale) soils and parcel (1:4,800 scale) data can be reasonably maintained as township coverages. These coverages are about 300K and 200K respectively. This storage approach is similar to the tiling scheme supported by the ARC/INFO map library (Aronson and Morehouse 1983). Such a structure has the potential to alleviate problems in handling regional scale-related variation within the same feature type. Dane County is bisected by a terminal moraine and soil geomorphologies across this boundary are quite different. Soil patterns within a township, however, are much more uniform, so generalization of this smaller unit is potentially simpler.

Resistence to storing a data base in separate units is due in part to past difficulties of merging them to create larger areal coverages. New software referred to as 'zipping' (Beard and Chrisman 1986) can overcome this difficulty. Given small areal coverage storage units as building blocks, the logical processing sequence is to generalize each unit to a desired level of detail. Then once the storage bulk is reduced the units can be "zipped" together quite rapidly to create smaller scale, larger areal coverages. To be workable, the combined generalization and "zipping" process for several units should be faster and less complex than generalization of one large detailed coverage.

Software development

The last part of the solution is to develop a flexible package of generalization tools. A change of scale is not available at the push of a button as some systems would promise. Reduction of detailed coverages to substantially coarser resolution requires more sophisticated processing than simple line generalization. Tomlinson and Boyle (1981) conducted a benchmark of nine geographic information systems in 1980 and reported that, "No sytem demonstrated a capacity to produce legible 1:250,000 scale maps from 1:12,500 scale source material". Automated scale changing capabilities have not improved substantially since then. An automated scale reduction process described by Leberl, Olson and Lichtner (1985) handles a reduction of 1:24,00 to 1:50,000 scale (about a factor of 2). Monmonier (1983) calls for operational algorithms that must handle scale reductions exceeding a factor of four. Reliance on a single detailed coverage could require scale reductions exceeding a factor of 25. Much of the current generalization effort is still focused on line generalization research (Zoraster *et al* 1984) which is becoming redundant and avoids the more difficult problems of scale reduction. To effectively handle large reductions a process should allow for variable reduction of line detail, feature simplification, and attribute reclassification and aggregation. The following section discusses automated generalization capabilities needed to carry out large scale reductions.

COMPONENTS OF AN AUTOMATED GENERALIZATION PACKAGE

Some people assume only three basic algorithms; simplification, smoothing and displacement are required, and that these should be designed to replicate the manual generalization process as closely as possible. Rhind (1973) was one of the first to recognize that automated generalization should not be a direct translation of manual techniques. Rhind (1973) identified the essential automated generalization functions as line sinuosity reduction, feature transposition, within and between category amalgamation, feature or category elimination, and graphic coding change. For 1973 this was an astute selection of functions which recognized the need to deal with attribute as well as spatial information. Graphic coding change is the only unnecessary function. Graphic represenration is a consideration in the generalization process but it should remain a separate function. Brassel (1985) offered a model for automated generalization that consisted of objects and functions; the main objects being points, lines, and areas, with eight to twelve functions for each of these objects. His model also recognizes the need for more complex functions specifically related to digital representations. In an attempt to be exhaustive, however, his model is perhaps more complex than necessary. Five major functions would appear to be optimal and are described below.

SELECT: This routine allows a user to select features for elimination or to select or exclude a set of features for further processing. Selection criteria can be geometric data such as threshold lengths, widths, areas, distances, perimeters, etc. or attribute data such as names or geocodes In some situations a desired generalization could be accomplished by the selection process alone. The result of this selection process should be a new topologically consistent coverage.

AGGREGATE: This process condenses the attribute information by reducing the number of classes. The user specifies new classes and defines how they will be hierarchically reclassified from existing classes of the detailed coverage. As an example, residential, commercial and manufacturing land uses might be aggregated to a built-up or urban land use class. This routine eliminated lines and areas based on the reclassification. In the land use example, any lines separating residential and commercial land uses will disappear, and the combined area will appear as an urban land use area.

REDUCE: This is a routine to remove points from a line. The Douglas-Peucker routine is a logical choice for this function. It works on the principle that points of maximum deviation from a trend line connecting the end points of a line are retained while points less than a specified distance (tolerance) from the trend line are eliminated.. This routine has a well defined theoretical base (Peucker 1975, Marino, 1979, White, 1983), is computationally efficient and mimimizes positional displacement (McMasters 1986). Its utility has been proven by the test of time and the adoption by nearly everyone.

COLLAPSE: This routine explicitly invokes a dimension change. Areas or polygons specifically selected can be collapsed to lines or points, i.e. a river represented as an area is collapsed to a single line, or a city represented as an area is collapsed to a point. Nodes are relocated on the center line or centroid. Neighboring areas are extended to occupy areas formerly covered by the collapsed feature.

COARSEN: This routine will simplify and also collapse features. The degree of simplification will depend on a specified distance, epsilon. This idea is based on the epsilon filter (Perkal, 1965, Chrisman 1983). The routine removes or modifies features by analyzing clusters of points which fall within epsilon of each other. As an example, if points defining an island or penisula are within this distance of each other, they will collapse to points and then be eliminated.

These functions can be invoked in different combinations and order to create the desired result. The number of steps and processing sequence depends on the final objective, scale and graphic output resolution. A large reduction to scale might require all of the functions or several iterations of a function. A number of these functions already exist in some form in such GIS as ODYSSEY and ARC/INFO (which implies that these require topology). With slight modification these functions could be adapted for use in an automated scale reduction package. The collapse routine does not yet exist but should not be extremely difficult to implement. The coarsen routine exists currently in ODYSSEY but needs refinement.

APPLICATION OF THE PROCEDURE

As a test of the single detailed database concept, this generalization procedure was applied to a detailed coverage to produce smaller scale versions. While not all of the generalization procedure is fully operational, the basic concepts were demonstrated on a simple data set. The data set was 1:10,000 to 1:40,000 scale digital coastline data provided by the National Ocean Service (NOS). The NOS data exemplifies the detailed coverage concept. Their database was constructed by digitizing only the largest scale, most detailed version of the coastline. Where 1:10,000 or 1:15,000 scale versions of the coastline were available these were substituted for the 1:40,000 scale coastline.

The success of the procedure was to be tested by comparison of the generalized results against an independently generalized product. The test data was provided by NOS and consisted of the detailed coastline data plotted, manually generalized to 1:250,000 scale and redigitized. To match the test data set, the detailed coastline data had to be reduced by a factor of 25, (a reasonably large scale reduction).

The NOS data was iteratively reduced and coarsened by using progressively larger tolerances. Table 1 summarizes this process. The final entry in this table is the manually generalized test data set.

	Number of polygons	Number of points
NOS detailed source data	65	2703
20 meter reduction	65	1362
20 meter coarsening	51	1291
50 meter coarsening	37	1013
50 meter reduction	37	597
80 meter coarsening	24	517
70 meter reduction	24	458
NOS manually generalized data	26	466

Table 1.

Maps 1 and 2 show the manually and automatically generalized versions respectively. As the number of points, polygons and graphic representations show, the automatically generalized version is a reasonable facsimile of the manually produced version. One of the main differences between the versions is the treatment of small islands. In the automated process these are eliminated, while in the manual generalization they are exaggerated. One advantage of the automated process is a reduction in error. Map 3 is an overlay of the manually generalized and original versions which shows the positional and attribute differences between the two. Map 4 is a similar overlay which shows the differences between the automatically generalized and original versions. Table 2 and a comparison of Maps 3 and 4 show that the both positional and attribute error in the automatically generalized result are reduced.

NOS Detailed Data Against NOS Generalized Data

		Hand Generalized Coastline	
		Land	Water
Detailed Coastline	Land	3928.18	313.25
	Water	101.73	4746.26

NOS Detailed Data Against Epsilon Generalized Data

		Automated Generalized Coastline	
		Land	Water
Detailed Coastline	Land	4144.93	90.76
	Water	109.25	4731.93

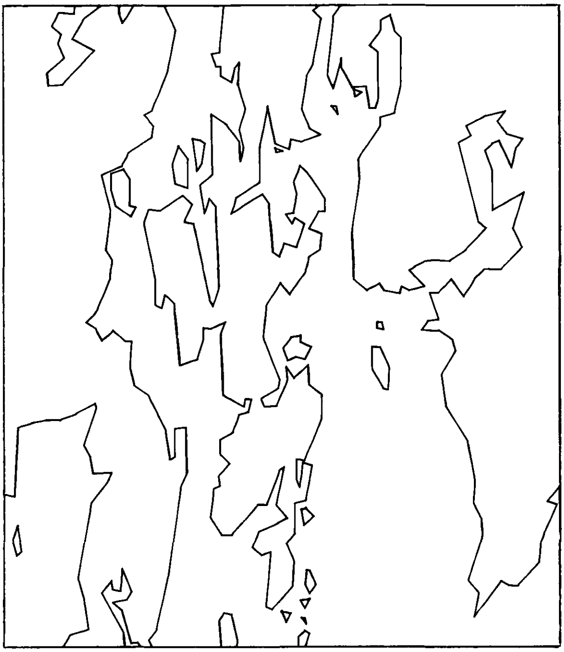
Table 2.

Another aspect of an automatically generalized product which cartographers have been striving for is a more objective result. The handling of features in this case was entirely objective. The only subjective input is a tolerance for the reduction and an epsilon distance for the coarsening.

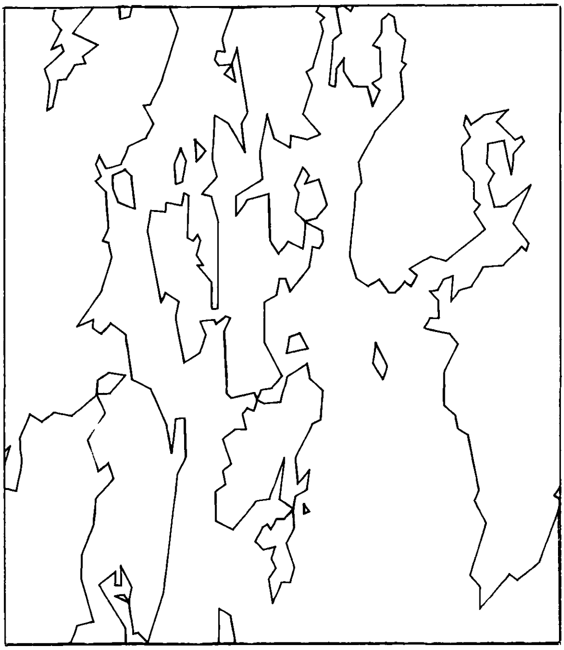
This test data involved the simple case of just two attribute classes; land and water, but the results are encouraging. The results show very reasonable reduction of the geometric data and handling of the range of different scales in the source material. With additional development and refinement of the functions, the process should be able to handle coverages with more complex attribute data.

ACKNOWLEDGEMENTS

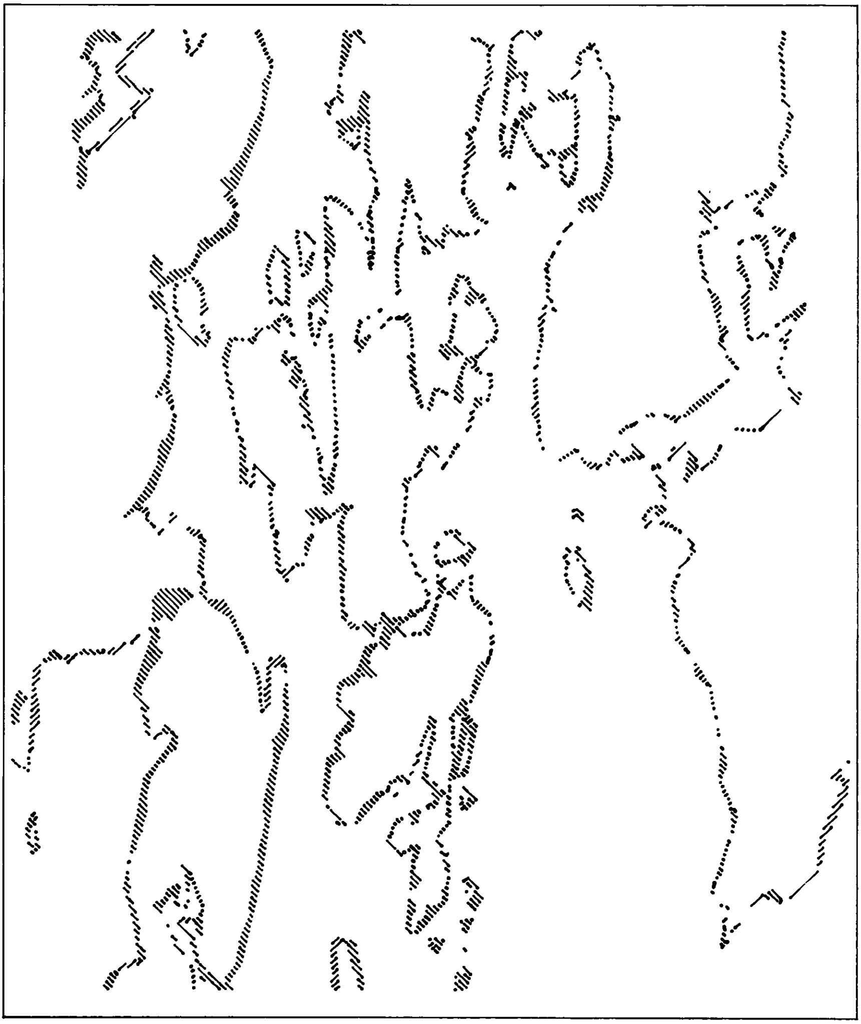
This research was funded in part by the Nautical Charting Branch of the National Ocean Service. The author thanks NOS for providing the digital coastline data.



Map 1



Map 2

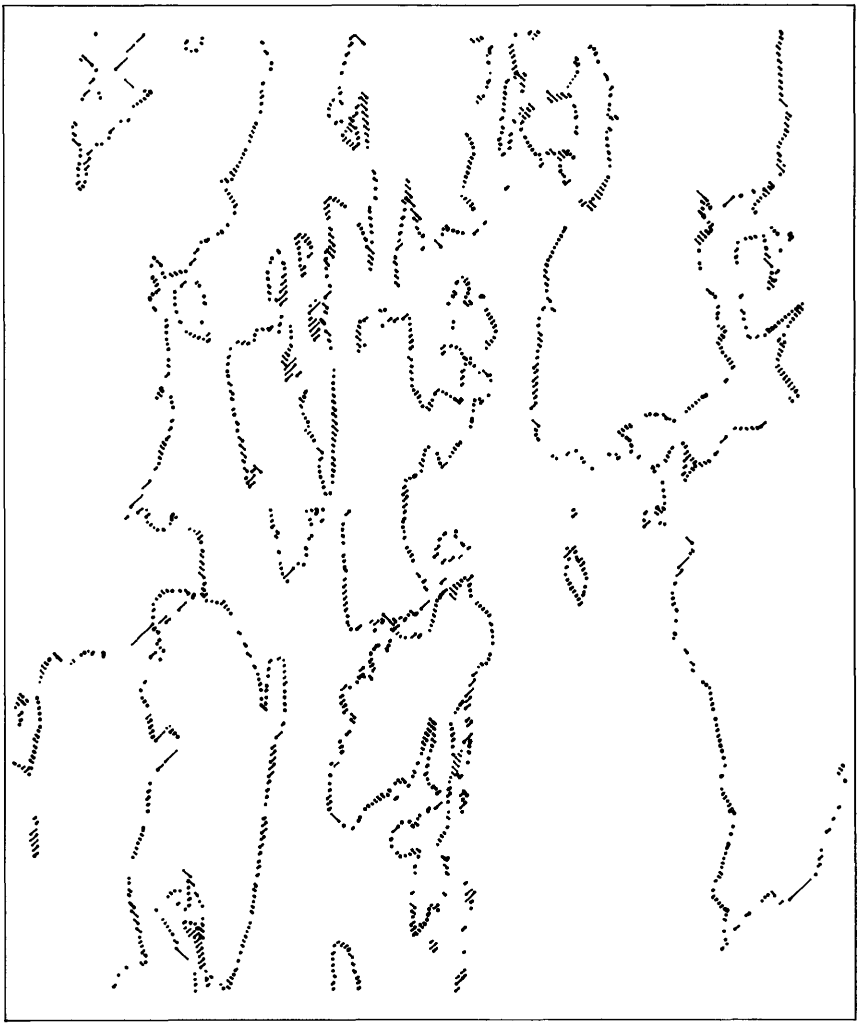


Map 3

Differences Between Large and Small Scale Charts
Generalized Coast Manually Derived from the Detailed Data

▨ Land in Large Scale, Water in Small

▧ Water in Large Scale, Land in Small



Map 4

Differences Between Large and Small Scale Charts
Coast Automatically Generalized from the Detailed Data

▨ Land in Large Scale, Water in Small

▧ Water in Large Scale, Land in Small

REFERENCES

Aronson P. and Morehouse, S., 1983. "The ARC/INFO Map Library: A Design for a Digital Geographic Database", **Proceedings AUTO-CARTO 6**, 1, 372-382.

Bauer, K., 1983. "Public Planning: the Role of Maps and the Geodetic Base", IES Report 123. University of Wisconsin-Madison. 130-140.

Beard, M. K., and Chrisman, N. R., 1986. "Zipping: New Software for Merging Map Sheets", **Proceeding of American Congress on Surveying and Mapping 46th Annual Meeting,** 1, 153-161.

Brassel, Kurt, 1985. "Strategies and Data Models for Computer-Aided Generalization", **International Yearbook of Cartography**, 25, 11-30.

Chrisman, N.R., 1982. Methods of Spatial Analysis Based on Error in Categorical Maps. unpublished Ph.D. thesis, University of Bristol.

Chrisman, N. R. 1983, "Epsilon Filtering: a Concept for Automated Scale Changing", **Proceedings of American Congress on Surveying and Mapping, 43rd Annual Meeting**, 322-331.

Chrisman, N.R. and Niemann, B. J., 1985, "Alternate Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning", **Proceedings AUTO-CARTO 7**, 84-94.

Jones, C.B. and Abraham, I. M., 1986, "Design Considerations for a Scale-Independent Cartographic Database", **Proceedings: 2nd International Symposium on Spatial Data Handling**, Seattle, WA 384-398.

Leberl, F., Olson, D., and Lichtner, W., 1985. "ASTRA: A System for Automated Scale Transition", **Technical Papers 51st Annual Meeting ASP**, 1, 1-13.

Marino, J., 1979. "Identification of Characteristic Points Along Naturally Occuring Lines: an empirical study", **The Canadian Cartographer**, 16, 1, 70-80.

McMasters, R.B., 1986, "A Statistical Analysis of Mathematical Measures for Linear Simplification", **The American Cartographer**, 13, 2, 103-116.

Perkal, J., 1966. "An Attempt at Objective Generalization', **Discussion Paper 10**, Ann Arbor, Michigan, Inter-University Community of Mathematical Geographers.

Peucker, T. K., 1975. "A Theory of the Cartographic Line", **Proceedings AUTO-CARTO 2**, 508-518.

Rhind, D.W., 1973, "Generalization and Realism Within Automated Cartography", **The Canadian Cartographer**, 10, 51-62.

Tomlinson, R., and Boyle, R., 1981. "The State of Development of Systems for Handling Natural Resource Data", **Cartographica**, 18, 4, 65-95.

Tomlinson Associates. 1986, Review of North American Experience of Current and Potential Uses of Geographic Information Systems, Tomlinson Associates, Ontario, Canada.

Watson, W., 1970. "A Study of the Generalization of a Small Scale Map Series", **International Yearbook of Cartography**, 16, 24-32.

White, E.R., 1983. "Assessment of Line Generalization Algorithms Using Characteristic Points", **The American Cartographer**, 12, 1, 17-27.

Zoraster, S., Davis, D., Hugus, M., 1984. **Manual and Automated Line Generalization and Feature Displacement** ETL-0359. Fort Belvoir, VA.

OPTIMUM POINT DENSITY AND COMPACTION RATES FOR THE REPRESENTATION OF GEOGRAPHIC LINES

J.-C. Muller
University of Alberta
Edmonton, Alberta, T6G 2H4

ABSTRACT

Previous work on automated line generalization has concentrated on the issues of computer efficiency for line thinning and the choice of a minimum number of critical points which must be retained in the generalized line. This study takes a different view of the problem of line generalization, emphasizing densification, rather than reduction, of the number of points describing a line for the purpose of optimizing representational accuracy. This new perspective raises the two following questions: 1) For a given map scale, what is the maximum number of describing points which can be retained without producing redundant information and 2) What is the relationship between line compaction rates and map scale reductions.

Given a plotter pen size, one may identify the smallest geographic artefact which may be visually recognizable. This, in turn, determines the minimum spacing of the digitized points describing the line. Finally, the concept of fractal dimension may be used to predict the maximum number of describing points for a given map scale, assuming statistical self-similarity for the geographic line.

The function governing the relationship between coordinate compaction rate and map scale reduction is particularly useful for the development of scale independent data bases, assuming that the points selected for small scale representations are always a subset of those used in larger scale representations. A linear relationship has been proposed but preliminary results show that the coordinate compaction rates depend on the generalization algorithm being used, the fractal dimension of the line as well as the map scale reduction.

INTRODUCTION

Not long ago, a consulting firm was commissioned by the US Army Engineer Topographic Laboratories to perform an extensive review of automated methods for line generalization (Zycor, 1984). The results of the study were inconclusive, and gave only broad recommendations as to which generalization algorithms appear potentially more promising than others. The study, among others published in the last few years, reflects the urgency of formalizing the process of cartographic generalization so that it can be adequately automated (Jenks, 1979; McMaster, 1983; White, 1985).

Previous work on the evaluation of automated line generalization has concentrated on the issues of computer efficiency and the choice of a minimum number of critical points which must be retained while preserving the geometric and visual characteristics of a geographic line. This paper proposes a different perspective on the problem of line generalization by emphasizing the idea of densification, rather than reduction, of the number of points describing the line with the purpose of optimizing geometric accuracy. Because storage costs and computing time for line storage and line thinning will become less and less of an issue, I take the view that the original line information should be preserved as much as possible, given the physical and conceptual limitations of map scale and map purposes. If we equate maximizing geometric accuracy to maximizing information --- in this case the number of points describing the curve -- then maximizing information implies minimizing data losses resulting from selection, displacement, simplification and deletion. This new perspective raises the two following questions: 1) For a given line and map scale, what is the maximum number of describing points which the map is able to bear, and 2) what is the relationship between line compaction rates and map scale reductions. The second question is raised in the context of a cartographic data base which is used to produce maps at different scales. It would be useful to know the relationship between scale and information compression since this relation could be implemented to determine automatically the number of describing points that should be kept to draw a line at a given scale. A further issue, of course, is the choice of the describing points which ought to be included. Although the focus of this paper is not the evaluation of generalization methods which most characteristically reproduce the geometry and the geography of a coast line or river, the method adopted for line generalization is a necessary consideration in this research. Specifically, the relationship between data compaction rates and map scales may be seriously affected by the method adopted for compressing the data.

PREDICTING THE OPTIMUM NUMBER OF DESCRIBING POINTS

The digital representation of a cartographic line usually takes the form of a discrete set of points identified by their positions with respect to an arbitrary coordinate system. Successive points are joined by linear elements to make up the line. Thus, this discrete set of points constitutes the core information about the line. One important practical question for the cartographer is how large must this set be in order to ensure a proper description of the line. If one wants an accurate representation, then the number of describing points must be as large as possible for the particular scale at which the line is to be plotted. This, in turn, implies that the points made available in the cartographic data base have been digitized with a resolution equal or higher than the corresponding resolution of the plot.

Let the line width of the plotter pen be 0.2 mm. By definition, open meanders whose width or wavelength is smaller than or equal to 0.2 mm cannot be drawn open (Figure 1). This would also be the case for a raster plotter whose resolution is less than or equal to 127 dots per inch. Assume further that the separation between two line strokes may not be smaller than 0.2 mm. This is a conservative estimate to ensure visual separability, although I am aware that visual acuity or the corresponding angle of visual discrimination measured in minutes of arc may allow much finer separations, depending on the reading conditions. Thus, only meanders whose wavelength is equal or larger than 0.4 mm can be drawn clearly and distinguishably (Figure 1). As a result, the describing points of the line may not be closer than Δ = 0.4 mm from each other. This means that digitizing a map in a stream mode with a distance between digitized points less than 0.4 mm would generate superfluous points. It also means that any further reduction

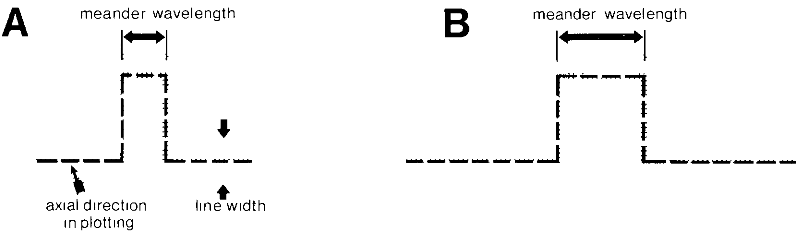


Figure 1. In A, the meander is closed in because its wavelength is no greater than the line width. In B, the meander is shown distinguishably. Its wavelength is no smaller than the line width plus the minimum separation allowed between line strokes.

of the original map to reproduce the line on a smaller map format implies a generalization such that no two consecutive points retained on the generalized curve are closer to each other than a distance $\Delta = 0.4$ mm. For instance, a three-fold reduction would imply the systematic elimination of one of the points which is member of a pair of consecutive points whose distance is smaller than 0.4 x 3 = 1.2 mm on the original copy. The nth point or distance-traversed generalization algorithm could be used to enforce such a condition when the selected points are to be a subset of the original points. Otherwise the walking generalization algorithm is a good candidate for the application of the minimum separation rule (Muller, 1987). It produces a new sequence of points which are equally distant from each other. In either case, collinear points may be subsequently removed in order to reduce the storage space taken by the results.

The above rule provides a guideline for maximum point density. It would be useful to predict the total number of describing points resulting from its application. The concept of fractal dimension may be used to calculate this number.

Assume the geographic line is a fractal, that is, each piece of its shape is geometrically similar to the whole. This property is called self-similarity (Mandelbrot, 1982). From Richardson, we have the equation (Richardson, 1961):

$$L(\varepsilon) = \varepsilon**(1-D) \tag{1}$$

where ε is the step length to measure the length of the line $L(\varepsilon)$, and D is a constant.

Let N be the number of steps ε used to measure the line length. Then $L(\varepsilon) = N \times \varepsilon$. According to (1):

$$N \times \varepsilon = \varepsilon**(1-D)$$

$$\ln N + \ln\varepsilon = (1-D)\ln\varepsilon$$

$$\ln N / \ln\varepsilon = -D$$

or

$$D = \ln N / \ln(1/\varepsilon) \tag{2}$$

D is called the fractal dimension of the line. For all practical purposes, the value $1/\varepsilon$ may be thought of as the number of steps of length ε partitioning the base line (a straight line joining the first and last point of the curve's basic fractal generator, which, in the case of a geographic line, is the whole line itself).

Note that equation (1) can also be rewritten:

$$D = 1 - \ln L(\varepsilon) / \ln(\varepsilon) \tag{3}$$

A geographic line is said to be statistically self-similar when the relationship between $\ln L(\varepsilon)$ and $\ln(\varepsilon)$ is linear. In this case, the limit $(\ln L(\varepsilon + \Delta\varepsilon) - \ln L(\varepsilon))/\Delta\varepsilon$ where $\Delta\varepsilon \rightarrow 0$, is estimated through regression analysis and is used to determine the fractal dimension in equation (3).

Furthermore, given the fractal dimension of a geographic line, one can determine the value of N:

$$\ln N = D \text{ x } \ln(1/\varepsilon)$$

or

$$N = e{*}{*}\ln(1/\varepsilon) \text{ x } D \tag{4}$$

The steps of length ε are the strokes which are used to draw the curve and, according to the minimum separation rule, may not be smaller than Δ, the minimum distance between the describing points of the curve. Assume again $\Delta = 0.4$ mm. One can calculate the value of N to predict the maximum number of points which may be used to describe the line, depending on its fractal dimension and the size of the plot (Table 1).

TABLE 1. MAXIMUM NUMBER OF DESCRIBING POINTS (N) DEPENDING ON FRACTAL DIMENSION (D) AND PLOT SIZE

PLOT SIZE = 40 mm;	Δ = 0.4 mm;	ε = 0.4/40;	1/ε = 100
D		N	
1.0		100	
1.1		158	
1.2		251	
1.6		1,585	
2.0		10,000	
PLOT SIZE = 160 mm;	Δ = 0.4 mm;	ε = 0.4/160;	1/ε = 400
D		N	
1.0		400	
1.1		728	
1.2		1,326	
1.6		14,564	
2.0		160,000	

Note: Plot size is calculated according to [(Xn - X1)**2 + (Yn-Y1)**2]**½ where (X1, Y1) and (Xn, Yn) designate the positions of the first and last points of the curve.

For illustration, the coast line of Banks in British Columbia was plotted on a pen plotter using the above approach, with pen size = 0.2 mm, Δ = 0.4 mm, and N = 1226 points for a fractal dimension of 1.2145 and a plot size of fourteen centimeters (Figure 2).

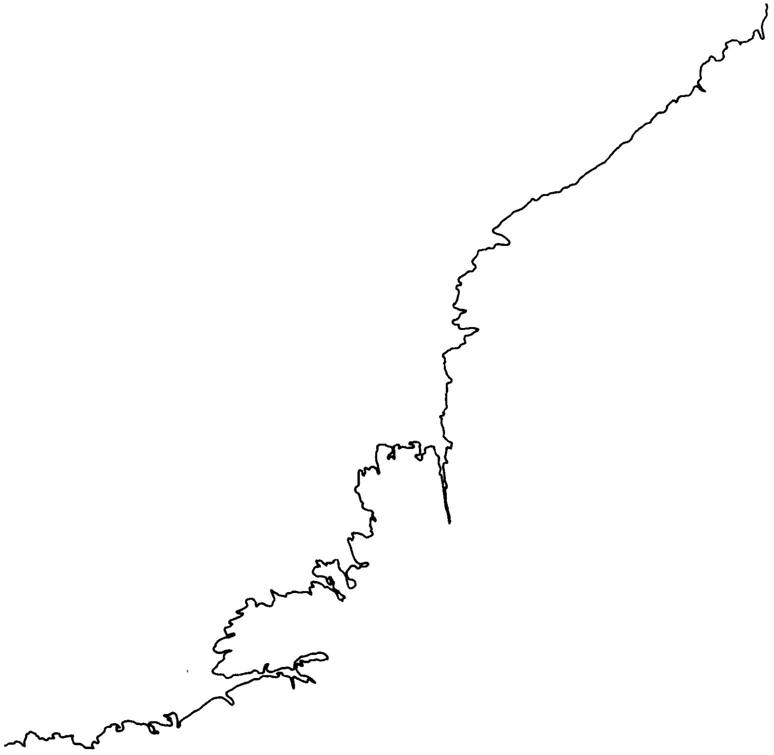


Figure 2. The Banks coastline with 1226 points.

The original Banks coastline was digitized on a 1/50000 topographic map from the NTS series 103 G/8. The number of digitized points was originally 2759 and was reduced to 1266 using the walking algorithm and a walking step Δ = 0.4 mm. Since the Banks coastline is self-similar (Muller, 1986), the value of N could be predicted for any plot size. A plot size of seven centimeters, for instance, would require only 528 describing points. The visual appearance of the plot in Figure 2 may still be unsatisfactory to the reader, as many spikes and meanders appear closed in. This phenomenon is not directly related to the size of the walking steps, however, but to the morphology of the original line which shows many narrow spikes and inlets which are almost circular (Figure 3). The spike problem has already been mentioned elsewhere (Deveau, 1985). Complex lines with narrow spikes and wide but circular meanders have a tendency to collide with themselves through the process of generalization. This is particularly true when a recursive tolerance band algorithm, such as the one of Douglas and Peucker (1973), is applied. A possible solution to this problem would be to identify all line segments which are crossing over, colliding or potentially colliding (within a particular tolerance window) and displace the corresponding points. Research is currently in progress in this area.

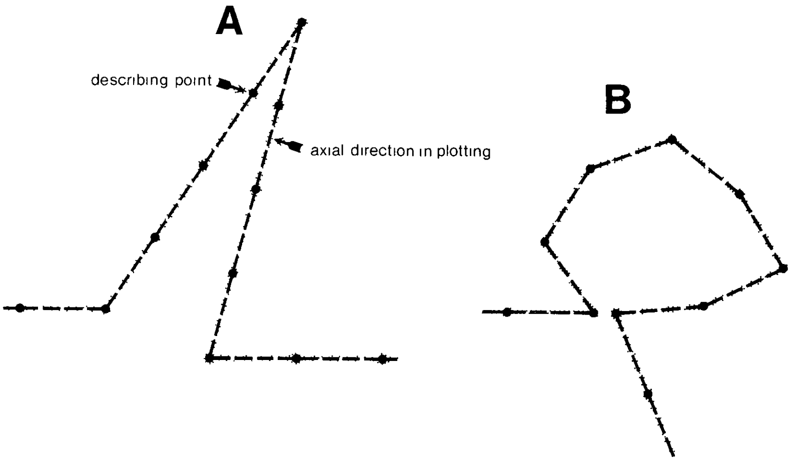


Figure 3. The spike (A) and the inlet (B) are partly or completely closed in.

Note that equation (4) provides the lower and upper limits of the number of points necessary to describe a line corresponding the lower (D=1) and upper (D=2) limits of fractal dimension (Table 1). When D tends toward a value of 2, the line tends to fill the space and thus requires a large number of points to be described (160,000 points for a point sampling resolution of 0.4 mm and a 16 cm plot size). The lower limit corresponds to a fractal dimension of D=1 which characterizes continuous, differentiable curves such as a circle or a straight line. In the later case, a blind application of the proposed approach would be absurd, since a straight line only requires two points and any other point included according to the minimum separation rule would be redundant. Again, a check on collinearity for any straight segment of the line would remove this problem. Another limitation of the proposed approach is the fact that N can be predicted for self-similar lines only. Previous studies have shown that geographic lines are not always self-similar (Hakanson, 1978; Goodchild, 1980).

RELATING DATA COMPRESSION RATES TO MAP SCALE

The relationship between scale and the quantity of information displayed on a map has been studied for quite some time. Several models to formalize this relationship have been proposed, among those the Radical Law or Principle of Selection by Topfer and Pillewizer (1966). For linear information, such as the data describing a geographic line, the Radical Law takes the simplest form:

$$N \times M = \text{constant} \qquad (5)$$

when N would be the number of points describing the line and M the denominator of the map scale. Accordingly, a two-fold reduction of the original map translates into a two-fold reduction of the number of describing points. Renewed interest was recently expressed for this type of empirical rule, as it "introduces the possibility of a hierarchical method of line storage, whereby the number of points retrieved is directly related to the scale of the required map". (Jones and Abraham, 1986). This hierarchical structure, however, implies that the points selected for small scale representations are always a subset of those used in larger scale representations, which is not always the case.

The issue here is whether the Radical Law, which proposes a linear relationship between map scale reduction and data compression for line data, has practical value. An empirical test was conducted on two coastlines -- one complex line (Banks, already mentioned) and one simpler line (Isidro, digitized from the 1/50,000, G12-B11 Gulf of California map). They were tested at four different scales, each scale being successively a two-fold reduction of the previous one. Three different generalization algorithms, including the moving average, the Douglas-Peucker and the walking algorithm, were used to represent the lines at the different scales, with generalization rates corresponding to the scale reduction rates (Figure 4 and 5). For the sake of

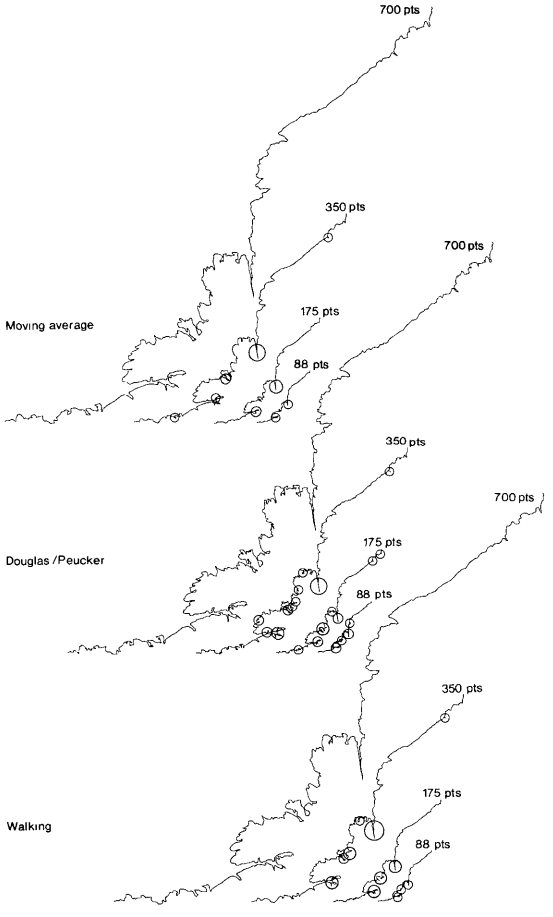


Figure 4. Scale Reduction and Corresponding Data Compression Using Different Generalization Algorithms on Banks Coastline. Problem areas in the smaller scale representations are highlighted by circles.

clarity, the number of describing points on the largest representations was purposely reduced in order to minimize the risk of line collision and afford a better comparison with the smaller scale representations. For all the Banks tests, the smaller scale representations show new problem areas (closing spikes and closed loops) or a worsening of the ones already present on the larger maps (Figure 4). Note, however, that the test using the Douglas algorithm gives the worst result. This suggests that the Radical Law is less suited for this generalization algorithm than for the others. The Isidro tests, on the other hand, were all successful, demonstrating that the Radical Law is applicable for simpler lines (Figure 5). This small experiment shows that the form of the relationship between data compression and scale reduction of linear elements is more complex than the one suggested by Topfer and Pillewizer and is a function dependent on line complexity and method of generalization as well. In the case of statistically self-similar geographic lines, one could incorporate the effect of

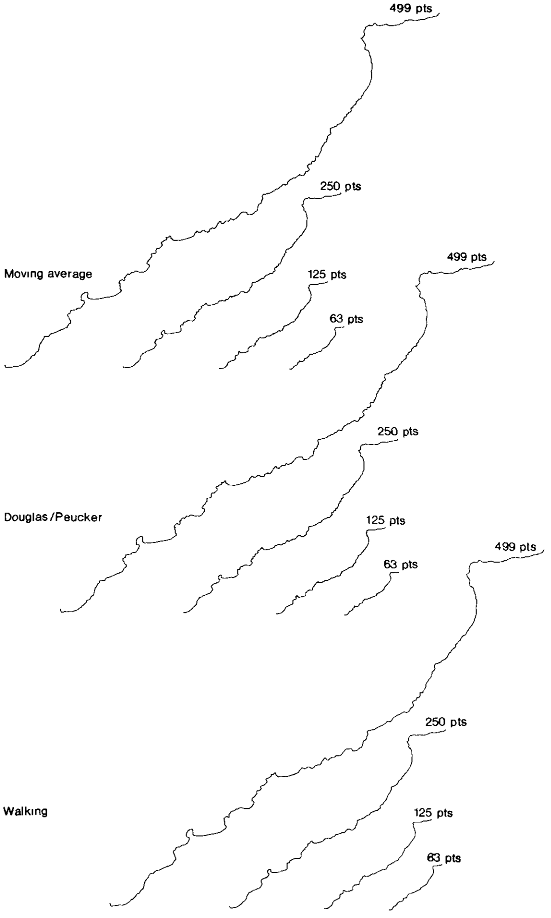


Figure 5. Scale Reduction and Corresponding Data Compression Using Different Generalization Algorithms for the Isidro Coastline.

complexity by suggesting the following relation:

$$N1 = NO\ [(MO/M1)**D] \tag{6}$$

where D is the fractal dimension of the line, NO and N1 are the number of describing points on the larger and the smaller scale maps, MO and M1 are the corresponding scale denominators. In the case of a space filling curve, the reduction in the number of describing points would correspond to the reduction in map area:

$$N1 = NO\ [(MO/M1)**2] \tag{7}$$

Although this relationship may be more suited for complex curves, its successful application depends upon the assumption of an appropriate point density on the original source map.

Furthermore, one could incorporate the minimum separation rule in equation (5):

$$\Delta 1 = \Delta 0\ [(M1/MO)**D \tag{8}$$

when $\Delta 0$ and $\Delta 1$ are the minimum spacing between the describing points on the original map and the new derived map after reduction. This would provide a rule for generalization as well as optimize point density for any particular scale. It could be easily applied in a hierarchical data base where the original describing points on the source document were captured through stream mode digitizing with a constant $\Delta 0$ value. The points selected in a smaller scale representation would be a subset of the original describing points according to the new minimum separation value $\Delta 1$.

CONCLUSION

A few guidelines for consideration prior to the process of line generalization have been proposed. There is the view that one ought to maximize the number of points describing the line for any particular scale. A minimum separation rule between describing points may be set as a function of plotting resolution and visual discrimination. For statistically self-similar geographic lines, the total number of points required to describe the curve according to the minimum separation rule may be predicted. The walking algorithm was applied to illustrate this rule. Furthermore, it was found that the Topfer and Pillewizer's Radical Law which suggests a linear relationship between data compaction and map scale reduction was not suited for complex lines. In the case of the self-similar lines, a relationship including the fractal dimension was proposed instead. A problem which deserves further investigation is the tendency of a complex curve to collide with itself through the process of generalization. A purely algorithmic solution to the problem of line generalization does not appear satisfactory. Cartographic generalization is not only a reduction of the amount of information for the sake of preserving map readibility (Salichtchev, 1977). Generalization also involves an understanding of the meaning of the information which is being generalized. Thus, there is a need to add some "intelligence" to the computer generalization process to insure that line segments are not colliding (topological integrity) and that significant geographic features are preserved (geographical integrity,) as would be case if the line was generalized by a cartographer using his geographic knowledge.

ACKNOWLEDGMENTS

The author wishes to thank Luca Vanzella for his technical help and the Cartographic Laboratory of the University of Alberta for the graphic reproductions.

REFERENCES

Deveau, T.J. 1985. "Reducing the Number of Points in a Plane Curve Representation," Proceedings, AUTO CARTO VII, 152-160.

Douglas, D.H. and T.K. Peucker, 1973, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature", Canadian Cartographer, Vol. 10, 110-122.

Goodchild, M.F. 1980, "Fractals and the Accuracy of Geographical Measures", Mathematical Geology, Vol. 12, 85-98.

Hakanson, Lars, 1978, "The Length of Closed Geomorphic Lines," Mathematical Geology, Vol. 10, 141-167.

Jenks, G.F. 1979, "Thoughts on Line Generalization," Proceedings, AUTO CARTO IV, 209-221.

Jones C.B. and I.M. Abraham, 1986. "Design Considerations for a Scale-Independent Cartographic Database," Proceedings, Second International Symposium on Spatial Data Handling, 384-398.

Mandelbrot, B.B. 1982, "The Fractal Geometry of Nature, Freeman, San Francisco.

McMaster, R.B. 1983. "Mathematical Measures for the Evaluation of Simplified Lines on Maps," Unpublished Ph.D. Dissertation thesis, University of Kansas, 333 pages.

Muller, J.C. 1986, "Fractal Dimension and Inconsistencies in Cartographic Line Representations," The Cartographic Journal, Vol. 23, in press.

Muller J.-C. 1987, "Fractals and Automated Line Generalization", The Cartographic Journal, Vol. 24, in press.

Richardson, L.F. 1961. "The Problem of Contiguity: An Appendix of Statistics of Deadly Quarrels," General Systems Yearbook, Vol. 6, 139-187.

Topfer, F. and W. Pillewizer, 1966. "The Principles of Selection," The Cartographic Journal, Vol. 3, 10-16.

Salichtchev, K.A. 1977, "Some Reflections on the Subject and Method of Cartography after the Sixth International Cartographic Conference," Cartographica, Monograph No. 19, 111-116.

Zycor, Inc. 1984, Manuals and Automated Line Generalization and Feature Displacement, Report for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060 U.S.A., 204 pages.

THE PSI-S PLOT:
A USEFUL REPRESENTATION FOR DIGITAL CARTOGRAPHIC LINES

Michael P. O'Neill*
Department of Geography
Virginia Polytechnical Institute and State University
Blacksburg, Virginia

and

David M. Mark*
Department of Geography
State University of New York at Buffalo
Buffalo, New York 14260

ABSTRACT

The Psi-s plot represents the geometry of a line by determining Psi, the orientation angle or heading, at points along the line, and plotting that value against s, the cumulative curvilinear distance along the line. No matter how convoluted the line is, the Psi-s plot will be a single-valued function of s; it is much easier to parameterize such a single-valued function. Straight lines on the Psi-s plot represent either straight lines or arcs of circles in x-y space. The Psi-s representation can be used to characterize the shapes of irregular polygons. It also has been used in analyses of river meander planform, and in the automated detection of contour crenulations. The Psi-s plot should be valuable in the generalization of digital cartographic lines; it should have an advantage over standard methods for representing geographic lines which include substantial proportions of circular arcs or straight lines. Circular arcs and straight lines are common components of rivers, roads, and railroads. The technique would appear to have great potential for feature recognition and shape characterization of digital cartographic lines.

* Electronic mail (BITNET) addresses for authors: ONEILLMP@VTVM1, GEODMM@UBVMS

INTRODUCTION

In any application of computers, the choice of an appropriate **representation** for the phenomenon being studied is crucial to algorithm design and in fact to the definition of problems themselves. Winston (1984, p. 21) defined two key concepts in artificial intelligence:

> "A **representation** is a set of conventions about how to describe a class of things. A **description** makes use of the conventions of a representation to describe some particular thing."

Winston then went on to emphasize the role of an appropriate representation in problem-solving. There has, however, been little attention in computer cartography to representations of geographic and/or cartographic lines. With few exceptions, cartographers have adopted the representation of a digital line as an ordered set of two or more coordinate pairs (a **chain**, or **polyline** representation).

One apparent exception to this lack of explicit attention to representation is Poiker's **Theory of the Cartographic Line** (Peucker, 1975), which attaches an explicit "width" parameter (W) to a line, and then represents the line as a set of "bands" or rectangular boxes, with widths no greater than W, which enclose the line. In fact, this "theory" was developed **a posteriori** to describe the model underlying the so-called "Douglas-Peucker" line reduction algorithm (Ramer, 1972; Douglas and Peucker, 1973) and related line-handling algorithms.

Recently, Buttonfield (1985) reviewed representations of cartographic lines and their variability. In addition to an overview of Poiker's model and the fractal model of line variability, Buttonfield briefly discussed a parameterization of a cartographic line, in which the x-and y-coordinates of successive points along the line are plotted against s, the accummulated distance along the line (Buttonfield, 1985, pp. 3-4). Although this approach simplifies the line, it produces two curves. Furthermore, equations fitted to either of these curves have no clear geometric interpretations.

In this paper, we present an alternative parameterization of a cartographic line. This **Psi-s curve** is an effective representation for cartographic lines, because straightforward geometric interpretations of the form of the transformed line are possible. The Psi-s curve appears to have potential both for line generalization and for pattern recognition in cartographic lines.

THE PSI-S PLOT

The Psi-s plot represents the geometry of a line by determining Psi, the orientation angle or heading, at points along the line, and plotting that value against s, the cumulative curvilinear distance along the line (see Figures

1 to 4, below). One advantage of this transformation is that, no matter how convoluted the line is, the Psi-s plot will be a single-valued function of s; it is much easier to parameterize such a single-valued function. Another advantage is that the shape of the Psi-s plot has a straight-forward geometric interpretation. Straight lines on the Psi-s plot represent arcs of circles in x-y space; in fact, if Psi is measured in radians, the radius of curvature of the arc is exactly the inverse of the slope of the Psi-s curve. Straight lines in x-y space can be considered to be arcs of circles of infinite radius, and appear on the Psi-s plot as horizontal straight lines (zero slope).

The Psi-s plot has been used in computational vision as an aid in characterizing the shapes of irregular polygons representing the outlines of machine parts (cf. Ballard and Brown, 1982). However, it has a longer history of use in geomorphology. Speight (1965) applied this transformation to river meanders when conducting a power spectral analysis of meander form. Then, Langbein and Leopold (1966) discussed this representation for meanders, pointing out that straight lines in Psi-s space represent circular arcs in x-y space. Brice (1974a, 1974b) extended this approach, claiming that a 'circles-and-straights' geometry (straight lines on the Psi-s curve) is more closely characteristic of meanders than are trigonometric functions (sine-waves in Psi-s). More recently, Mark (1986) applied the method to the automated detection of contour crenulations, and O'Neill (1987) extended the use of Psi-s plots for characterizing the planform geometry of river meanders.

PSI-S PLOTS OF GEOGRAPHIC LINES: SOME GENERAL EXAMPLES

The concept of the Psi-s plot and its relation to geometry in geographic (x-y) space can be made clear through the presentation of examples. In this section, we present maps and Psi-s plots of segments of three geographical lines: a meandering river (Figure 1), a winding mountain highway (Figure 2), and a contour line (Figure 3).

A Meandering River. The Hay River in northern Alberta provides an good example of a wandering stream with "free" (i.e., relatively unconstrained) meanders. The segment of the Hay River going downstream from X to Y (Figure 1, above) is represented on the Psi-s plot (Figure 1, below). Letters show the correspondence between bends or straight reaches on the river and straight line segments on the Psi-s plot. Note that sharp bend (such as the one at "g" on the Hay River) plot as very steep segments on the Psi-s plot, whereas more open bends (eg., "c") have lower slopes. Bend "c" in fact appears to be a compound curve, with a more open middle component (lower slope segment on the Psi-s plot) between sharper entrances and exits from the bend. There are no straight segments in this reach.

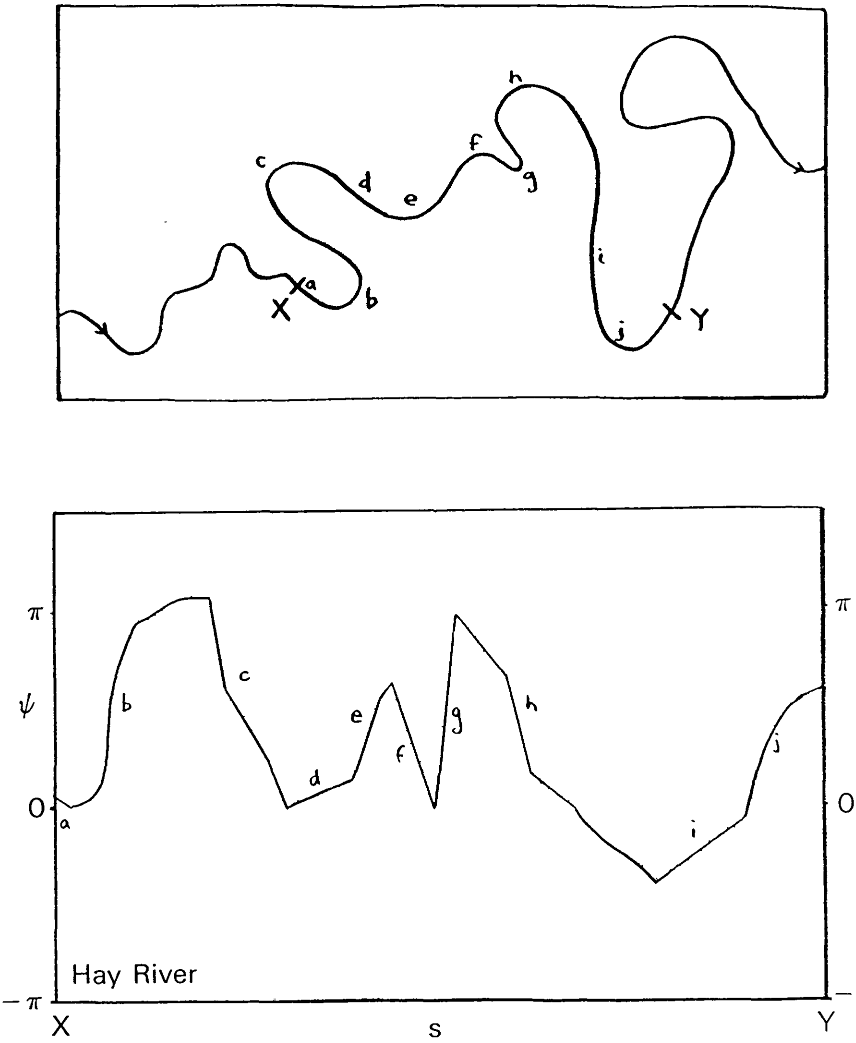


Figure 1: A portion of Hay River, Alberta (above) and its Psi-s plot (below). Equivalent parts of the two plots are marked with lower case letters. Straight lines on the Psi-s plot represent straight lines if horizontal, and circular arcs otherwise.

A Mountain Highway. We noted above that roads were another type of geographic line which should be well-suited to the Psi-s representation. As in the case of rivers, roads commonly are composed of relatively straight segments joined by approximately-circular bends. Here, there a few straight segments, but most of the highway is composed of curves. Although the two lines are very similar in x-y space (cf. Mark, 1985, Figure 2 E and K, p. 49), there is a fairly distinct difference in the Psi-s plots. Note that the geographic length of a segment is its extent along the s axis; thus a visual examination of the Psi-s plot tends to exaggerate the importance of bends.

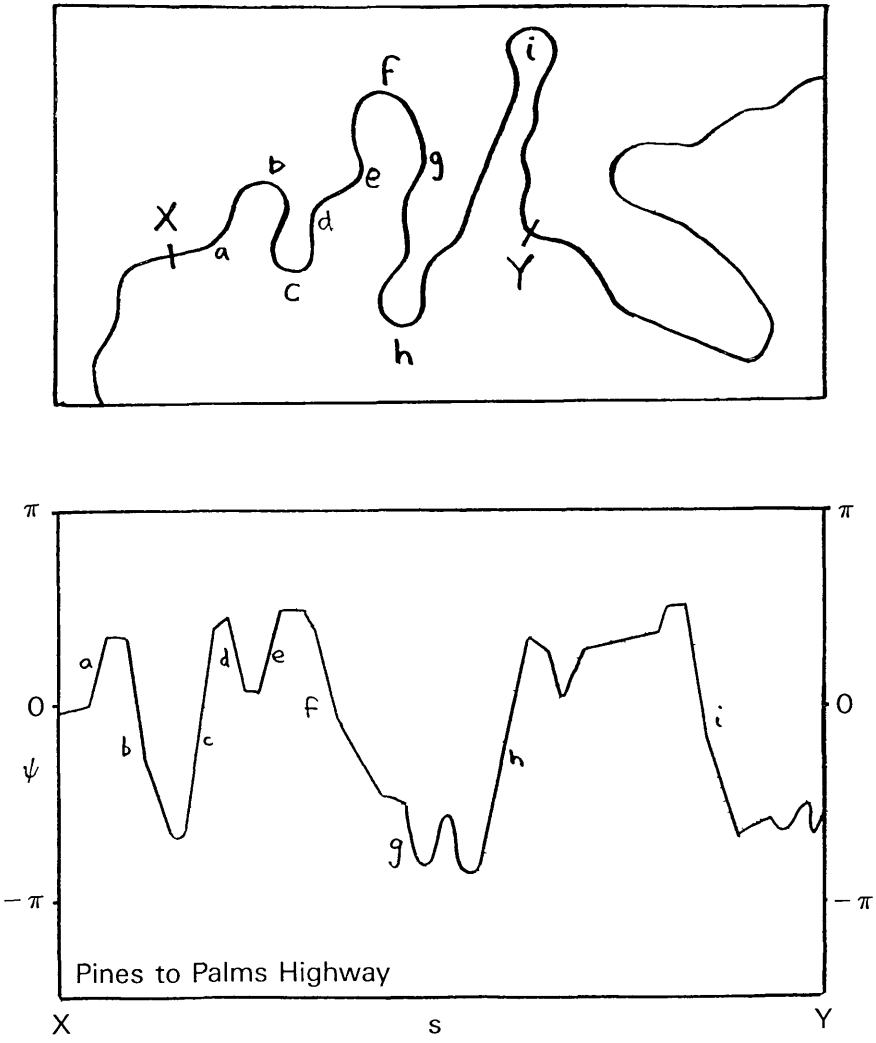


Figure 2: A portion of the Pines to Palms Highway, southwest of Palm Springs, California (above) and its Psi-s plot (below).

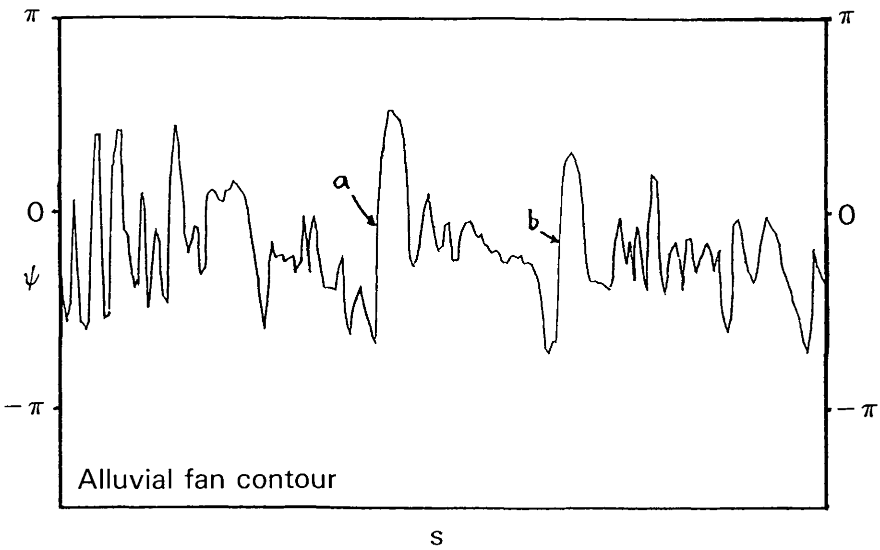


Figure 3: Psi-s plot of a portion of the 4800 foot contour on an alluvial fan in Arizona. Two contour crenulations (see text below) are marked at "a" and "b".

A Contour Line. Contours and shorelines are geographical lines which are not usually composed of circular arcs. (Exceptions might be found in cirques and other glacial landforms, or along spits.) Thus, it probably will take as many straight line segments to represent the contour on the Psi-s plot as it would on the x-y map (for an equivalent level of generalization). This impression is support by the irregular appearance of the Psi-s plot, which is presented at about the same scale as the other lines.

Whereas the Psi-s transformation may not reduce storage requirements in this case, it can be very valuable in the identification of features on contour lines (Mark, 1986). This will be discussed in more detail below.

THE PSI-S PLOT AND RIVER MEANDER RESEARCH

In their classic theoretical paper on river meander form, Langbein and Leopold (1966) developed a solution to the path of a river which, while fitting a total channel length L into a straight line distance $D < L$, minimized the variance of angular change along the path. Their probability analysis was most easily developed when stream azimuth was expressed as a function of distance along the channel (in other words, on a Psi-s plot). They found that the angle-change variance was minimized when the plot was a sine function (in Psi-s space), corresponding with what they called a "sine-generated curve" in cartesian coordinates (Langbein and Leopold, 1966, p. H3).

Langbein and Leopold went on to note that sine-generated curves have "a relatively uniform radius of curvature in the bend portion" (Langbein and Leopold, p. H3), which corresponds with "the fact that a sine curve has quite a straight segment as it crosses the x-axis" (Langbein and Leopold, p. H3). This straight segment represents about 1/3 of the meander length. Also, they noted that the slope of the Psi-s curve "is the reciprocal of the local radius of curvature of the meander" (Langbein and Leopold, p. H3). They constructed Psi-s plots of several natural rivers and one meander trace from a flume study, and then fitted sine-curves to portions of these, apparently 'by eye'. These curves fit rather well, but it should be noted that fits were over just one or two bends, and the reaches selected were mostly 'well-known' meanders. Langbein and Leopold did not report fitting straight lines to the Psi-s plots.

Brice (1974a, p. 582) credited Langbein and Leopold (1966) with having devised the technique of plotting azimuth (Psi) against distance as measured along the stream (s). However, Brice himself clarified the relation between planform geometry and straight segments of the Psi-s plot, stating that "segments of the resulting plot, which have a uniform slope, represent arcs of uniform curvature" (Brice, 1974a, p. 582). Brice claimed that arcs of constant curvature are more common in natural channels than the 1/3 proportion suggested by Leopold and Langbein (1966, p. H4) for equilibrium channels. Brice fitted line segments to portions of the Psi-s plot "by eye" (p. 586) to identify segments of constant curvature. In another paper published the same year, Brice (1974b) restated the description of his method, and emphasized the fact that horizontal segments of the Psi-s curve represent straight lines in cartesian space (Brice, 1974b, p. 185).

PSI-S PLOTS AND THE ANALYSIS OF DIGITAL CARTOGRAPHIC LINES

One major advantage of using a Psi-s representation is that when digitized points are equally-spaced along the cartographic line (spacing = ds), the line is completely characterized by a vector of direction (Psi) values. Clearly, the storage of a cartographic line is greatly reduced if direction (Psi) measurements are stored as a vector with known spacing of s rather than a paired set of x-y coordinates. Additionally, techniques of line generalization can be applied to the Psi-s representation to further reduce storage requirements of the line.

Line Generalization Using the Psi-s Curve.

O'Neill (1987) used the Psi-s representation in a new approach to cartographic line generalization. Just as a complicated cartographic line can be approximated (to some specified accuracy) by a series of straight line segments in the plane (Douglas and Peucker, 1973), so can such a line be approximated by straight line segments on the Psi-s plot. Note that this is equivalent to approximating the line by a sequence of straight lines and circular arcs in cartesian space. A slightly modified version of the Douglas-Poiker line generalization algorithm (Douglas and

Peucker, 1973) can be used here; the modification is needed because the axes of the Psi-s plot are not dimensionally homogeneous.

One solution is to measure deviations from a straight line in a direction parallel to the Psi-axis. If all points over some range of s lie within the specified tolerance, then the segment can be represented as a circular arc; otherwise, the point of maximum deviation is found, and the test is applied recursively to the two halves of the segment.

Psi-s Curves and Feature Identification for Cartographic Lines.

Recently, Mark (1986) discussed how the Psi-s representation can be used to identify contour crenulations in digitized versions of contour lines on pediments and alluvial fans. In the Psi-s representation, a contour crenulation appears as an abrupt change of almost 90

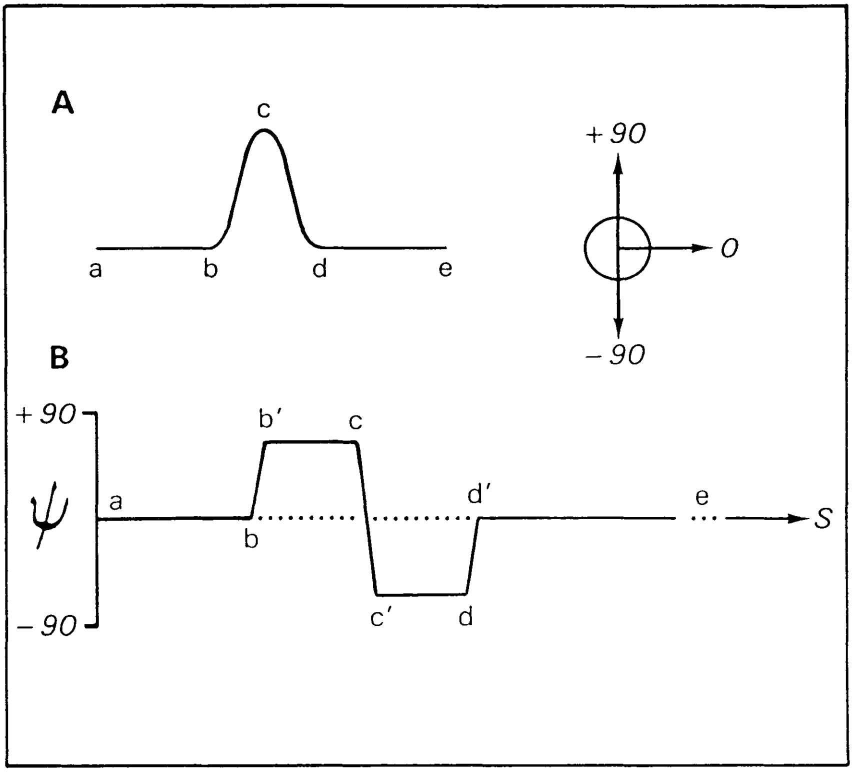


Figure 4: A portion of a contour crenulation (A) and the associated Psi-s plot (B). The place where the contour crosses the channel ("c") has the largest angular change on the Psi-s plot. [after Mark, 1986, p. 231].

degrees (pi) as the contour enters the small valley in which the channel lies ("b" in Figure 4), a change of almost 180 degrees (2 pi) in the opposite direction as the contour crosses the channel ("c" in Figure 4), and finally another 90-degree (pi) turn as the contour leaves the valley ("d" in Figure 4). The recognition of these features is facilitated through the calculation of the first differences of this series (departures from a straight line at each polyline vertex). Note that the near 180-degree change uniquely identifies the down-slope direction in this environment. Line segments "a" and "b" in Figure 3 represent two clearly-marked 180-degree direction changes at contour crenulations.

SUMMARY

The Psi-s curve has already proven to be a powerful representation for the analysis of river meander planform. It also has considerable potential in automated classification and feature recognition for digital cartographic lines. The use of the Psi-s curve should provide more compact yet effective generalizations of certain cartographic lines than can geometric generalizations in x-y space. Any geographic lines which include substantial proportions of circular arcs should be more effectively handled in the Psi-s representation, whereas the representation should be of little or no advantage for irregular lines (those which resemble fractals). Circular arcs are common components of rivers, roads, and railroads. Transformation to a Psi-s plot may not be worthwhile for coastlines and contours, unless feature recognition or shape characterization is the objective.

REFERENCES

Ballard, D. H., and Brown, C. M., 1982, Computer Vision: Englewood Cliffs, New Jersey, Prentice-Hall.

Brice, J. C., 1974a. Evolution of meander loops: Geological Society of America Bulletin, 85, 581-586.

Brice, J. C., 1974b. Meandering pattern of the White River in Indiana--An Analysis: in M. Morisawa (editor), Fluvial Geomorphology, State University of New York, Binghamton, pp. 178-200.

Buttonfield, B., 1985, Treatment of the cartographic line: Cartographica, 22, 1-26.

Douglas, D. H., and Peucker, T. K., 1973, Algorithms for the reduction of the number of points required to represent a digitized line line or its caricature: The Canadian Cartographer, 10, 112-122.

Langbein, W. B. and Leopold, L. B., 1966, River meanders--theory of minimum variance: United States Geological Survey Professional Paper 422-H, p. H1-H15.

Mark, D. M., 1985, Fundamental spatial patterns: The meander. Ontario Geography, 25, 41-53.

Mark, D. M., 1986, Knowledge-based approaches for contour-to-grid interpolation on desert pediments and similar surfaces of low relief: Proceedings, Second International Symposium on Spatial Data Handling, Seattle Washington, July 8-12, 225-234.

O'Neill, M. P., 1987, Meandering channel patterns -- analysis and interpretation: Unpublished Ph.D. dissertation, State University of New York at Buffalo, Buffalo, New York.

Peucker, T. K., 1975, A theory of the cartographic line: Proceedings, Auto-Carto II, 508-518.

Ramer, U., 1972, An iterative procedure for the polygonal approximation of plane curves: Computer Graphics and Image Processing, 1, 3.

Speight, J. G., 1965, Meander spectra of the Angabunga river: Journal of Hydrology, 3, 1-15.

PERFECTING AUTOMATIC LINE DRAWING

Kurt K. Kubik
Joseph C. Loon
Department of Geodetic Science and Surveying
The Ohio State University
Columbus, Ohio 43210-1247

ABSTRACT

New interpolation methods are presented, yielding results which are closer to manually interpolated lines than classical interpolation methods. By better modelling the line structure, these methods do not need the definition of breakpoints, the definition of which is necessary due to shortcomings in the classical methods.

CLASSICAL LINE DRAWING

In automated cartography it is often required to interpolate lines or surfaces in between given points. In one dimension this operation can readily be visualized: Given a set of points (x,z), find the interpolated value z at location x. Various methods are in use for this purpose. One may use local linear interpolation in between neighbouring control points, local polynomial interpolation, spline interpolation, prediction and moving average rules or other methods. These methods can usually be related to each other, as was shown by Frederiksen et al, 1984. Usually these methods use smooth interpolation, minimizing the average curvature of the line.

Take for example the cubic spline. It minimizes the function

$$I = \int_{\Omega} \left(\frac{\partial z}{\partial x}\right)^2 d\Omega \; ; \quad \Omega \text{ domain} \tag{1}$$

In order to understand the behavior of this mathematical spline, we may think in its physical analogue, the draftsman's spline. In drawing a smooth curve through a number of points, the draftsman uses a thin elastic rod forced to pass through the locations of the points. The shape of this mechanical spline is such that it has a minimum bending or curvature. Consider now points (x,z) in figure 1.1, representing a step function. Both the physical (draftsman) spline and the mathematical spline cannot properly describe the abrupt changes in height values and tend to oscillate heavily. In order to get reasonable results, one must introduce two "break points" where the tangent (slope) of the curve is discontinuous, and interpolate independently within the resulting three sections, figure 1.2.

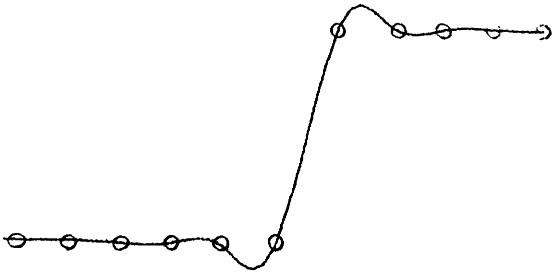


Figure 1.1: Spline interpolation of a step function

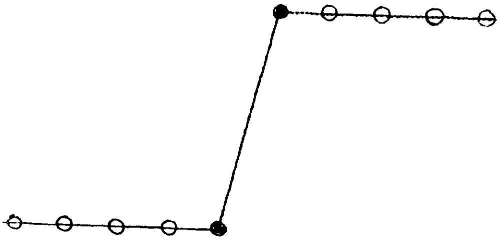


Figure 1.2: Spline interpolation with break points

Similar considerations are valid for other interpolation methods (finite elements, Prediction method, local interpolation).

The necessity of defining break points (or break lines in two-dimensional interpolation) may thus be seen as due to the deficiency of the interpolation methods to properly model the terrain. The economic consequences of this deficiency are considerable: in modern DEM (Digital Elevation Model) applications one spends on the average most of the time digitizing break lines in order to get an acceptable result (see figures 1.3 to 1.5, ref: Kubik, 1985). It thus pays off to investigate other interpolation methods which reduce the need of break point and break line definition.

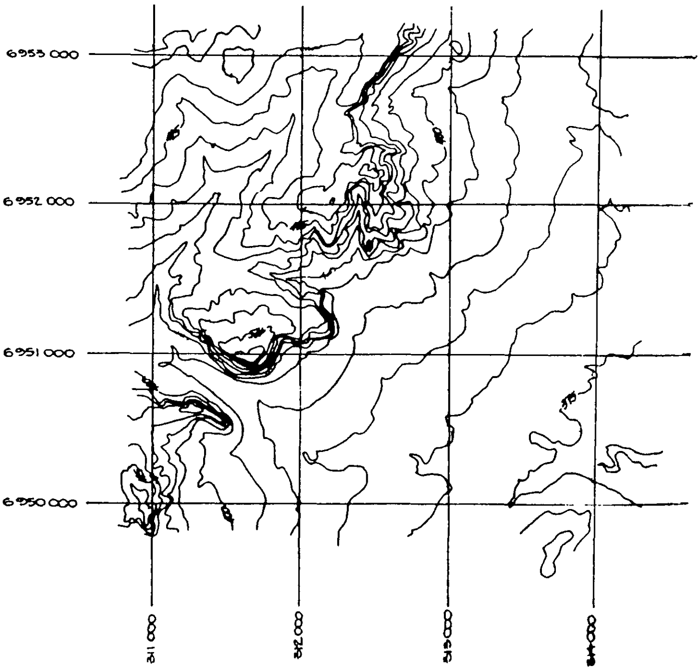


Figure 1.3: Contour lines of test area (5 meter contours) 7,964 points digitized, 90 minutes measuring time.

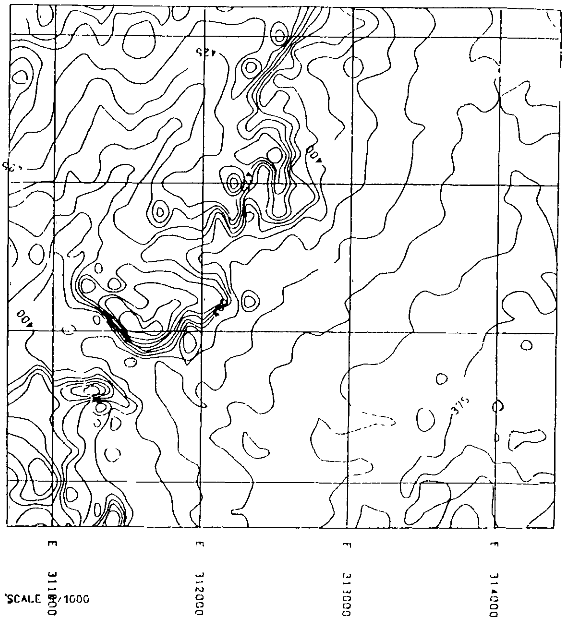


Figure 1.4: Results of HIFI profiles of 100m spacings; characteristic profile points selected by operator; no breaklines (1,162 points digitized, 43 minutes measuring time)

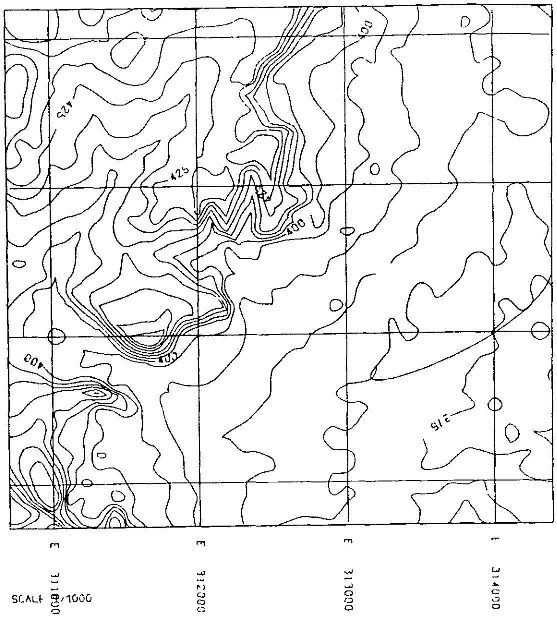


Figure 1.5: Results of HIFI profiles of 100m spacing operator selected points; breaklines included (1,121 profile points and 2,542 breaklines points digitized, 96 minutes measuring time)

NEW INTERPOLATION METHODS - MINIMIZING OTHER DERIVATIVES

As concluded in the earlier chapter, the classical interpolation programmes use relatively inflexible interpolation functions. By minimizing the average second derivative of the interpolation function, these methods (approximately) describe the bendings of a mechanical spline or plate, but not the undulations of a cartographic line. As an alternative, we may try to model the behavior of this line by the differential equation

$$z^{(n)} = \varepsilon \qquad (2)$$

where (n) denotes the n-th derivative of the values z, and ε is an independent random variable (white noise). Here n may also be non-integer. In that latter case this fractional n-th derivative is defined by continuous interpolation into the integer differences (Frederiksen et al, 1984).

This differential equation (2) may now be chosen as a model for interpolation, for instance in L-spline or finite element interpolation. The proper functional to be minimized is then

$$J = \int_{\Omega} \left\{ \frac{\partial^{(n)} z}{\partial x^{(n)}} \right\}^2 dx \rightarrow \min. \qquad (3)$$

The results of the L-spline interpolation are indentical to the results of the Wiener prediction method using a proper variogram or covariance function, as it was shown already in 1971 by Kimeldorf and Wabha (see also Dolph and Woodbury, 1952 and Kubik, 1973).

Figure 2.1 shows examples of interpolation according to these new principles. The digitized points represent the profile of a well known cartographer. For n = 1 we obtain piecewise linear interpolation (linear spline), for n = 2, piecewise 3rd degree interpolation (cubic splines) and for n between 1 and 2 we obtain interpolation forms which properly model break points in the terrain profile while preserving relative smoothness in the other profile sections. From extensive analysis of various terrain forms, the authors found n values in between 1.2 and 1.4 as most appropriate for use in DEM applications. Manmade cartographic lines are modelled on the average with a slightly larger n value. Algorithms for online determination of the proper n value for individual lines were developed by the authors to enable proper interpolation according to the line structure inherent in the digitized points (Kubik and Loon, 1985).

[1] The coefficient n can be derived from analysis of the cartographic line, using the variogram or spectrum concepts, see (Frederiksen et al, 1984).

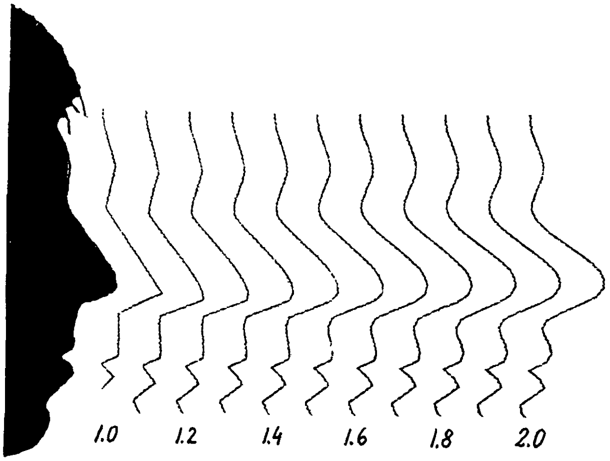


Figure 2.1: Interpolation with fractionals derivatives; use of different values of n for interpolation

NEW INTERPOLATION METHODS - MINIMIZING OTHER FUNCTIONALS

The above idea can be further generalized by minimizing well chosen functions of the derivatives, instead of their square sum:

$$J = \int_{\Omega} f\left[\frac{\partial^{(n)} z}{\partial x^{(n)}}\right] d\Omega \rightarrow \min.$$

In order to illustrate this principle, we consider the simple functionals

$$J_p = \int_{\Omega} \left|\frac{\partial^2 z}{\partial x^2}\right|^p d\Omega \quad ; \quad 0 \leqslant p \leqslant 2$$

minimizing the integral of non-integer power of the second derivative of the line.

The numerical interpolation methods for solving (5) are analogous to the methods described in Chapter 2. In order to demonstrate the effect of this class of interpolation principles, we choose again the profile of a well known cartographer (Figure 3.1). Classical cubic spline interpolation (using $p = 2$) yields unsatisfactory results and would need the definition of numerous break points to yield an acceptable result. 3.4 shows the interpolation results for decreasing values of p. Notice that the profile becomes more recognizable for decreasing values of p, with an optimal choice of p equal to 1.2.[2] Lower p values yield an increasingly rough profile, with piecewise linear interpolation obtained for $p = 1$.

[2] This optimal value of p can also be derived from covariance or spectral analysis of the sample points.

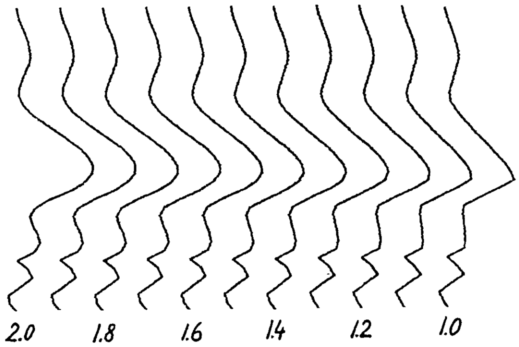


Figure 3.1: Interpolation with fractional powers; use of different values of P for interpolation.

Thus, with the principle (5), we have obtained a new transition of interpolation forms from a cubic spline to a linear spline, different from Chapter 2. In both cases, no break points were needed to yield realistic interpolations, which are close to the lines drawn by draftsmen.

Obviously, other functionals (4) may be chosen, which may give both worse and better interpolation results. However, proper use of these new principles allows a very effective ınterpolation of cartographic data, and considerable savings in data capture, as compared to classical methods.

FINAL REMARKS

Algorithms for rapid interpolation according to these new principles have been developed by the authors, and the methods were fine tuned for various classes of line interpolation (and approximation). As proposed in this paper, adaptive interpolation strategies are possible, taking into account the internal structure of the data set. However, much work still is necessary in order to fully understand the potentials of these new classes of interpolation methods. In particular, online data capture intertwined with interpolation appears desirable in order to help the operator in the digitization (or measuring) process and in understanding the nature of the interpolation function. This approach will allow a very considerable reduction of data capture time as compared to today's process.

REFERENCES

Frederiksen, P., Jacobi, O., Kubik, K. 1985, A Review of Current Trends in Terrain Modeling, ITC Journal 1985-2, plot 106

Kubik, K. 1985, A Digital Elevation Model for Queensland - Feasibility Study, Report of the Queensland Institute of Technology, Brisbane, Australia, 400 pp.

Kubik, K. and Loon, J. 1985, On Local Fractional Integration, Israeli Annals of Physics, 6 pp.

THE TIGER STRUCTURE

Christine Kinnear
Geography Division
U.S. Bureau of the Census
Washington, DC 20233

ABSTRACT

In an effort to automate the geographic support process for censuses and surveys, the United States Bureau of the Census has undertaken to design, develop, and implement an integrated geographic support system that will produce computer generated maps, assign addresses to geographic units and delineate geography for the collection, tabulation and publication of census data. The core of this effort is the Topologically Integrated Geographic Encoding and Referencing (TIGER) System, a collection of computer files and software. The central file of this system is the TIGER data base that houses the data required to perform the geographic support functions. This paper will describe the TIGER File structure, the elements that make up the file and the linkages between the elements.

OVERVIEW

The overall structure is shown in Figure 1. While it is one interlocking structure, it can be viewed as many smaller substructures, each performing a specific function. Four broad views are presented here: geometry, geography, linear attributes, and landmarks and related entities.

The TIGER File contains roads, hydrography, boundaries, railroads and some miscellaneous features that are represented as single lines. The location in space of the intersections of these lines is known to the TIGER file; and the areas enclosed by the intersecting lines are the atomic areal unit maintained in the file. These three units are known as 1-cells, 0-cells and 2-cells, respectively. They represent the geometry in the file and are related such that 1-cells are bounded by 0-cells and 2-cells are bounded by 1-cells. This relationship is the topological base upon which the remaining data are dependent.

The geographic elements are the areal cover and are linked to the topology via the 2-cells. There are three general sets of geographic combinations, made up to five specific components, that cover the 2-cells directly: the 1980 blocks and Geographic Tabulation Unit Base (GTUB), the 1990 blocks and GTUB, and the 1990 Ancillary GTUB. The 1980 data are kept for postcensus comparability studies. The Ancillary GTUB will be used to store the geographic entities not required for initial census tabulations, but needed for special postcensus tabulations. Individual geographic entities are linked to the three GTUBs. Each entity is made up of many GTUBs and each GTUB in turn is made up of many 2-cells.

Each 1-cell may be labeled with one or more feature identifier (a name), though it is expected it will be primarily roads and hydrography that will be named. Each feature identifier in turn may have one or more 1-cells attached to it. Each 1-cell/feature identifier combination may have one or more address range pair linked to it if the 1-cell is an addressable feature. The five-digit ZIP code is a linear attribute in the TIGER File and is linked directly to the address range pair. Therefore, the address range/ZIP code combination is linked indirectly to 1-cells.

The last view is the remaining miscellaneous items of Landmarks, Areas, and Key Geographic Locations (KGLs), collectively known as LAKs. Landmarks and KGLs are free floating points and are associated with 2-cells and 1-cell sides, respectively. Landmarks are cartographic enhancements, are linked directly to a 2-cell, and are generally unnamed. KGLs, mainly commercial buildings, are in the file as alternative addressing schemes and therefore are linked directly to an address range and subsequently to a 1-cell side; KGLs are always named. Areas are specially defined areas that are not part of the normal geographic hierarchy but are significant enough to warrant recognition, such as national parks and large water bodies. Areas consist of one or more 2-cells and may or may not be named.

The TIGER physical file is made up of many logical subfiles, each of which is generally one of the components shown on the structure diagram. Each subfile is made up of fixed length records that contain pointer data and descriptive data for the item represented. A subfile has data stored either ordered by a key (a balanced tree logical subfile), or randomly (Random Access Logical Subfile or RALS). Each subfile is linked to at least one other subfile. The balanced trees are the directories and provide the entry points into the data base.

The relationships in the TIGER file among the subfiles are explicitly represented by pointers. These relationships are specifically defined and are described as lists.

LISTS

An owner or head of a list is a single entity that is related potentially to many others. A member or tail is an entity that is in a subordinate relationship to another entity. Pointers are linkages between records and may be either a directory key or a subfile position number. (Unless otherwise noted, reference to pointers means position number.) There are three classes of pointers: 1) first or head to tail; 2) head/owner or tail to head; 3) next or tail to tail.

A list is a chain of like elements all of which have a common attribute. A list is effected by a combination of a first, head, and next pointers. It has an owner record on which is stored the common attribute and a first pointer to a tail record. The chained elements are the tail records that may have an owner pointer and may have a next pointer to another tail record.

There are five characteristics of lists in the TIGER File.

1. Simple: The owner has a first pointer and the tail may contain a next pointer. Simple lists can be one-way, where there is either a first or last pointer; or two-way where there are both.

2. Intertwined: A form of a simple list where the owner has a first pointer and the tail has a pair of next pointers, either of which may be used for the next record. Each tail is in two list of the same type simultaneously.

3. Many-to-many: This is a combination of two related simple lists each list owner having a first pointer to a shared relationship record. These tail records have next and owner pointers for each head.

4. Multilist: These are parallel simple lists where there are many owners with first pointers and the tail records contain parallel next and owner pointers.

5. Indexed: A simple list whose head is an ordered directory record with a first pointer and the tail may or may not contain a next pointer. Directory keys, rather than record numbers are used for head pointers.

Lists are the traversal mechanism in the TIGER File. Each is specifically defined in terms of its owners and its members. Referring to the structure diagram, lists are represented by the lines and arrows between subfiles. The double arrows indicate the next pointers of a list; or in other words, the subfile pointed to is a tail record of a list with next pointers. The single arrow on the other end of the double-arrow line represents the presence of an owner pointer on the tail record. A line with a single or double arrow at only one end represents a tail record without a pointer back to its owner. The many-to-many lists are represented on the structure diagram by the presence of a relationship subfile. Extending each owners double-arrow line through the relationship subfile to the other owner shows this relationship.

SUBFILES

The following discussion is a brief summary of the primary subfiles, including comments on the their relationships to other elements.

Ø-Cells

Ø-cells as the TIGER File views them are, in addition to points of intersection, terminating points of a line and intermediate points on a line that represent major attribute changes. Each Ø-cell is represented twice in the file, once on the Ø-cell directory in the form of a Peano key and once on the Ø-cell RALS in the form of longitude and latitude. Each Ø-cell is the owner of an intertwined list that has the 1-cells as tails.

A Peano key is a merge of the alternating bits of the coordinates of a point. It allows storage of the coordinates in a one-dimensional array.

2-Cells

2-cells are the atomic areal unit in the file. The TIGER File allows internal structures, both contiguous to and isolated from the bounding 1-cells, to be part of a 2-cell. The 2-cell RALS contains one record for each unique 2-cell. Though some data is stored on the record, the critical function of the 2-cell is the link between all geography and the topology. Each 2-cell is the owner of the intertwined list between 2-cells and 1-cells; and it is the tail on all GTUB and block related lists.

1-Cells

1-cells are the joining elements in the TIGER File. They link to feature identifiers, addresses, ZIP codes, KGLs; are bounded by Ø-cells; they bound 2-cells that link to all geography and landmarks and areas. The 1-cell RALS stores each unique 1-cell once, uniquely defined by the bounding Ø-cells plus curvature. Stored on the 1-cell record is an envelope, a minimum area that encloses the 1-cell plus its curvature. This is used to reduce spatial searches involving 1-cells. The remaining data consist mainly of boundary flags that identify the major geographic and statistical areas that the 1-cell bounds. The 1-cell is the tail of the two intertwined lists from the Ø-cells and 2-cells. It is these two list that represent the topological relationship in the TIGER File. A 1-cell may be the owner of one of the related simple lists in the many-to many relationship between 1-cells and feature identifiers. A 1-cell may also be the owner of the only two-way list in the TIGER File, the 1-cell to its curvature points.

1-Cell Curvature

1-cell curvature records can be perceived as an extension of the geometry of the 1-cell. Each 1-cell to curvature list contains all the coordinates for the intermediate points between the two Ø-cells. There is no limit to the number of curvature points a 1-cell may have. For plotting and computational convenience, the 1-cell to curvature list can be traversed both forward and backward.

Feature Identifiers and Feature Name Continuations

The feature identifiers are the labels of nongeographic entities, such as streets, rivers, buildings, and parks. Each unique identifier is represented only once in the feature identifier RALS even if it labels both a linear and an areal feature. Each identifier consists of at least a name and may include directions, such as east or northwest, and feature types, such as avenue, river, or building. There is a directory for feature identifiers and that directory is ordered by a Soundex key plus a truncated, packed version of the name. Therefore a directory entry may link to one or more feature identifier RALS records.

Since records are fixed length, a name continuation subfile is used to store name text that exceeded the fixed length. A feature identifier may be the owner of up to two related simple lists of two many-to-many lists: the feature identifier to 1-cell and the feature identifier to LAKs.

Address Ranges

Address ranges are stored as a left/right pair, with orientation being the same as the 1-cell they are associated with. Since address ranges are not linked directly to a 1-cell but to the combination of a feature identifier and 1-cell, each alternate identifier may have its own unique set of address ranges, allowing parallel address systems to be represented. The majority of the information on the address range RALS consists of flags that indicate the type of range, potential error conditions, and verified anomalies. The nine-digit ZIP code is considered part of the address range; the five-digit ZIP code being implicitly on the record as the ZIP code owner pointer of ZIP code to address range list. The four-digit add-on is stored explicitly as 1-cell side data. An address range may be the owner of a cross-reference address.

Cross-Reference Addresses

The Key Geographic Locations in the TIGER File are part of the overall automated address geocoding process and therefore need to be linked to a 1-cell side. Links to 1-cell side are via an address range; therefore, a KGL needs to have some link to an address range. The cross-reference address record serves this purpose with an actual or imputed address. A cross-reference address is the owner of an address to KGL list. This subfile is not intended for storage of the 199Ø census individual addresses.

ZIP Codes

ZIP codes in the TIGER File are looked upon as a linear or a point attribute. Each five-digit ZIP code that occurs in the file is stored once on the RALS record and once in the directory. The ZIP code RALS record is the owner of two lists, the ZIP code to address range and the one of the related simple list in the ZIP code to LAK many-to-many list.

Landmarks, Areas and Key Geographic Locations

This subfile is the collection of the three LAK entities described in the overview. There is one record for each unique LAK. Some information stored on the record is unique to specific entities, but flags indicate what usages are intended for each LAK. The LAK subfile may be the owner of up to three related simple lists of three many-to-many lists: LAK to feature identifier, LAK to ZIP code, and LAK to 2-cell.

Geographic Entities, Geographic Entity Extensions

Each of the 26 geographic entities recognized for decennial census processing is represented on the entity subfile; each entity is identified with its own record type. The data varies from entity to entity but each record contains as a minimum the combination of FIPS and Census codes required to uniquely identify the entity. An open ended entity extension subfile stores additional data for various entities. Currently record types exist on this subfile for areal data and entity names. Other record types may come into existence as required. The entity directory and RALS each have one record for each unique entity.

The entity subfile records are both the owners and the members in the many-to-many relationships among entity types. This allows an entity to be either dominant or subordinate to any other entity. The entity record is the owner in all the entity to GTUB multilists.

198Ø Geographic Tabulation Unit Base

Each unique combination of 198Ø tabulation geography is stored on both the 198Ø GTUB RALS and directory. Except for detected errors, the 198Ø GTUB will not be updated once it is associated with its 2-cells. Because of residual errors, some 2-cells may have no 198Ø GTUB owner and some GTUBs may have no 2-cell members.

199Ø Geographic Tabulation Unit Base

Each unique combination of 199Ø primary publication geography is stored on both the 199Ø GTUB RALS and directory. The geography will be current until the final update, at which time it will represent the final 199Ø geography. Each GTUB will be the owner of a 199Ø GTUB to 2-cell list.

199Ø Ancillary Geographic Tabulation Unit Base

The Ancillary GTUB RALS and directory contain one record for each combination of additional geographic codes. The geography on this subfile represents additional postcensus tabulation requirements. It also documents historical areas. Each GTUB will be the owner of an ancillary GTUB to 2-cell list.

198Ø Block/Enumeration District

This subfile contains the 198Ø block numbers and enumeration districts (ED). There are two record types, a block record and an ED record. There is no overlap of coverage; either a 2-cell was blocked in 198Ø and it therefore has only a block record; or it was not blocked and has an ED record. There is one record for each unique block code or enumeration district on both the RALS and the directory.

199Ø Block/Address Register Area, Block/ARA Extension

This subfile contains information for the smallest geographic area defined for the 199Ø census. The block RALS has two record types, a block group (first digit of a block number), and a block. The 2-cells initially are linked to the block group record, but once collection blocks are assigned, the link is to blocks only and block group records become summary data only. Collection blocks become tabulation blocks once splits by final 199Ø geography are effected. The Block/ARA extension subfile provides for additional data relating to blocks. Currently areal data is stored, but additional record types may be added if required.

The directory contains three record types, one for address register area (ARA), which is equivalent to a block group, but assigned a code unique within district office; one for block group; and one for block. Both the ARA and block group records link to the same block group record on the block RALS

Relationship Subfiles

This group of subfiles exists primarily to allow many-to-many relationships. This relationship shows two complementary views: each owner may have many members and each member may have many owners. Each one of these subfiles in the TIGER File has data that relates only to the combination of the two heads of the related simple lists. The 1-cell/feature identifier relationship record is the only one of this category that becomes an owner of another list. This occurs because address ranges are not associated directly with a 1-cell or a feature identifier, but rather with that 1-cell when it is known as that feature identifier. Therefore, this relationship record may become the owner of a list with address ranges as tail records.

The three relationship subfiles that are associated with the landmarks, areas, and key geographic locations are shown as separate entities, but they are in fact stored in one physical subfile with the data distinguishing one from the other.

SUMMARY

The data base described in this paper is designed to be resident on one of the Census Bureau's mainframe Unisys (Sperry) 1100 series computers. The software to manipulate this data base is written in FORTRAN-77 with Unisys supported extensions. The TIGER file will made up of many physical files partitioned by geographic units, mainly county or groups of counties.

The TIGER File structure described here is not the final structure, but is the TIGER-I(nterim) File to be used for the 1990 Decennial Census and the 1992 Economic and Agriculture Censuses. The TIGER System will continue to evolve throughout the next decade until it reaches its full expectations.

REFERENCE

U.S. Bureau of the Census, Geography Division, 1986, TIGER-I System Documentation.

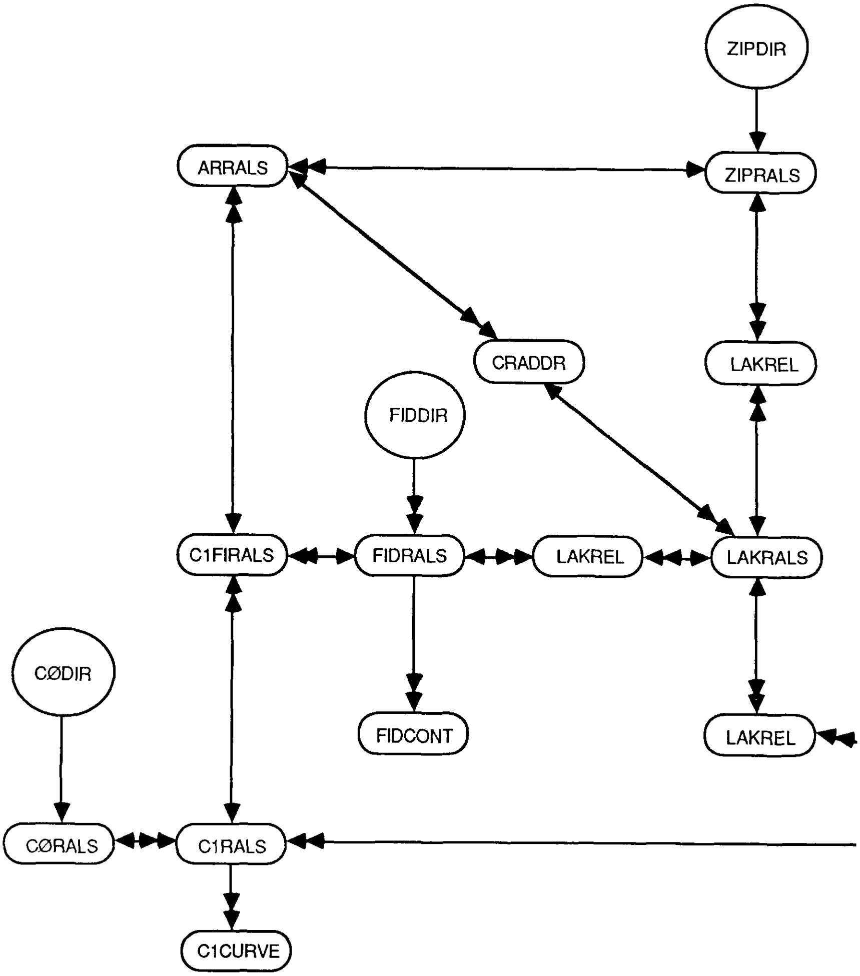


FIGURE 1. TIGER-I SYSTEM DATABASE STRUCTURE

KEY TO SYMBOLS:

▭ RANDOM ACCESS LOGICAL SUBFILE ○ BALANCED-TREE DIRECTORY

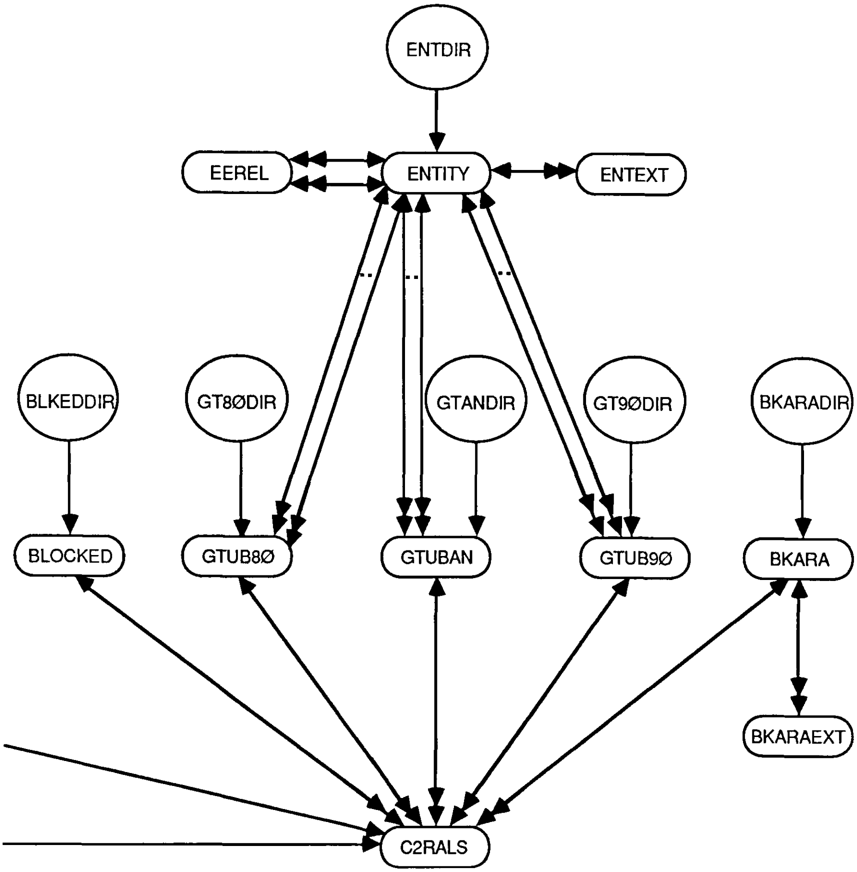
⟶ ONE-TO-ONE RELATIONSHIP ⟶▶▶ ONE-TO-MANY RELATIONSHIP

◀—▶▶ ONE-TO-MANY RELATIONSHIP WITH MEMBER POINTING BACK TO ITS OWNER

•• PARALLEL MULTIPLE LINKAGES

KEY TO SUBFILE ABBREVIATIONS:

ARRALS	- *ADDRESS RANGES*
BKARA	- 199Ø BLOCK/ADDRESS REGISTER AREA (ARA)
BKARADIR	- 199Ø BLOCK/ARA DIRECTORY
BKARAEXT	- 199Ø BLOCK/ARA EXTENSION
BLKEDDIR	- 198Ø BLOCK/ENUMERATION DISTRICT DIRECTORY
BLOCKED	- 198Ø BLOCK/ENUMERATION DISTRICT
CØDIR	- Ø-CELL DIRECTORY
CØRALS	- Ø-CELL
C1CURVE	- 1-CELL CURVATURE
C1FIRALS	- 1-CELL/FEATURE IDENTIFIER RELATIONSHIP



KEY TO SUBFILE ABBREVIATIONS (CONTINUED):

C1RALS - 1-CELL
C2RALS - 2-CELL
CRADDR - CROSS-REFERENCE ADDRESS
EEREL - ENTITY TO ENTITY RELATIONSHIP
ENTDIR - GEOGRAPHIC ENTITIES DIRECTORY
ENTEXT - GEOGRAPHIC ENTITIES EXTENSIONS
ENTITY - GEOGRAPHIC ENTITIES
FIDCONT - FEATURE NAME CONTINUATION
FIDDIR - FEATURE IDENTIFIER DIRECTORY
FIDRALS - FEATURE IDENTIFIER
GT8ØDIR - 198Ø GEOGRAPHIC TABULATION UNIT BASE DIRECTORY
GT9ØDIR - 199Ø GEOGRAPHIC TABULATION UNIT BASE DIRECTORY
GTANDIR - 199Ø ANCILLARY GEOGRAPHIC TABULATION UNIT BASE DIRECTORY
GTUB8Ø - 198Ø GEOGRAPHIC TABULATION UNIT BASE
GTUB9Ø - 199Ø GEOGRAPHIC TABULATION UNIT BASE
GTUBAN - 199Ø ANCILLARY GEOGRAPHIC TABULATION UNIT BASE
LAKRALS - LANDMARKS, AREAS, KEY GEOGRAPHIC LOCATIONS
LAKREL - LANDMARK, AREA, KEY GEOGRAPHIC LOCATION/ FEATURE IDENTIFIER, 2-CELL, ZIP CODE RELATIONSHIP
ZIPDIR - ZIP CODE DIRECTORY
ZIPRALS - ZIP CODE

TOPOLOGY IN THE TIGER FILE

Gerard Boudriault
Geography Division
U.S. Bureau of the Census
Washington, D.C. 20233

ABSTRACT

The topological structure in the Topologically Integrated Geographic Encoding and Referencing (TIGER) File is a realization of James Corbett's topological model of two-dimensional feature networks. The TIGER File's topology conforms to the model and adopts specific conventions. Software that maintains conformity to the model is essential.

INTRODUCTION

The Bureau of the Census is developing a geographic information system known as the TIGER System. The first two letters of the acronym stand for the words, Topologically Integrated. It is not for reasons of creating a clever name that Topologically Integrated is included. Topologically integrated is, in fact, the key to the power and beauty of the TIGER File. The topological structure contained in the TIGER File integrates geographic features, ZIP codes, address ranges and the myriad of political and statistical areas into a self-consistent Geographic Encoding and Referencing System. Without the integrating function of the TIGER File's topological structure, the geographic information needed to support the 1990 decennial census would likely be reduced to a disparate set of computer files and traditional maps.

The topological structure in the TIGER File is an application of the theoretical model of two-dimensional networks developed by James Corbett. Corbett's model uses the topological entities called 0, 1 and 2-cells to represent visible and non-visible features on the earth's surface. The model contains a set of topological rules in addition to the entities. This model is generally known in the automated cartography field. The TIGER File incorporates the basic theory of 0, 1 and 2-cells, but as a specific implementation of the theory, the TIGER File design adds unique rules and conventions to the theory.

Software designed for the TIGER File must be capable of maintaining the topological structure and rules or the TIGER File would become invalid. This essential requirement implies the construction of a software library whose algorithms and functions are determined by the model. Strict adherence to the rules is a prerequisite to the successful functioning of the TIGER System as a nationwide geographic information system.

TOPOLOGICAL ENTITIES

Corbett's topological model of features on the earth's surface is based on the representation of linear features, their intersections, and the areas they bound by the mathematical objects-- lines, points and areas respectively. The model specifically originates from the area of mathematics called combinatorial topology which uses the terms Ø-cell, 1-cell and 2-cell in place of point, line and area. Combinatorial topology delineates the sets of both global and local relationships that exist between these cells. The ability to derive the relationships between the cells means a corresponding ability to derive the relationships among the features that the cells represent. Hence the power and dynamic of the model.

1-cell

A 1-cell is defined as a sequence of at least two points where each point has a coordinate in the chosen coordinate space. In addition to the point sequence, the 1-cell implicitly contains the points between (in the sense of analytic geometry) adjacent points in the sequence. In parametric form a 1-cell consists of the set of points,

$$\begin{aligned}
&\{X|\ X=P_1, P_2, \ldots\ P_n \text{ where } P_i <> P_{i-1} \text{ and}\\
&X=P_1+(P_2-P_1)t,\\
&X=P_2+(P_3-P_2)t,\\
&\ldots\\
&X=P_{n-1}+(P_n-P_{n-1})t \text{ for } Ø<t<1\}.
\end{aligned}$$

The first and last points of the sequence are by definition Ø-cells, the first point is the "from" Ø-cell of the 1-cell and the last point is the "to" Ø-cell of the 1-cell. Every 1-cell is thereby bounded by its from and to Ø-cells. The points of a 1-cell other than its from and to Ø-cells are the curvature points and they serve to trace the shape of the feature identified with the 1-cell. Other terms commonly used for curvature points are shape points, curve points, and intermediate points. The TIGER File design imposes no limit on the number of curvature points that a 1-cell can possess.

A pair of adjacent points and the implied points between them comprise a line segment or vector. Using the concept of vector, a 1-cell may be conceived as a string or chain of vectors. The 1-cell definition's requirement that adjacent points are not equal excludes the legitimacy of a vector having a length of zero. By extension a 1-cell also may not have a length of zero. Zero length vectors and 1-cells are called degenerate and are forbidden in the TIGER File.

A 1-cell for which the from and to Ø-cells are the same is called a loop 1-cell. A loop 1-cell must possess at least two curvature points to form a simple closed curve. Loop 1-cells are legal in a TIGER File and are the most efficient means of representing unconnected lakes and so forth.

The 1-cell definition, it may be noted, does not mention 2-cells. At various points in TIGER File building it is valid to use a topological file whose sole topological entities are the zero-dimensional points and the one-dimensional lines with there being no set of areas coded to the sides of 1-cells. For these unpolygonized files, the Ø-cell and 1-cell entities are completely valid. (For example, the U.S. Geological Survey's 7.5-minute road overlay files were tagged without the benefit of 2-cells.)

Ø-cell

Ø-cells are zero-dimensional objects or points. In TIGER File topology, a Ø-cell is a point in the coordinate space and as such a Ø-cell possesses exactly one coordinate. The TIGER File's coordinate space is the spherical system of longitude and latitude. The coordinates are stored as fixed-point decimal values and have six digits of precision within a degree providing an average precision in absolute distance of 4 inches.

From the 1-cell definition it is known that the endpoints of 1-cells are Ø-cells. The TIGER File design also states the converse, namely that every Ø-cell is the endpoint of at least one 1-cell. This rule, in conjunction with the prohibition on degenerate 1-cells, disallows the use of Ø-cells to represent point features such as mountain peaks and radio towers unless the feature is coincident with a linear feature represented by a 1-cell. Note: The TIGER File design represents such free standing point features in a related landmark structure.

2-cell

2-cells are two-dimensional objects or finite regions of the coordinate space. A point is said to belong to a 2-cell if there is a two-dimensional sphere such that the point is its center and all of its interior points belong to the 2-cell. This ensures that 2-cells are continuous. The extent of a 2-cell is determined by the left and right sides of a set of 1-cells that is analogous to Ø-cells bounding 1-cells.

The complex of 1-cells in a TIGER File covers a finite region of the infinite projective plane. The unbounded region that surrounds the 1-cell complex is always represented by the 2-cell labelled with the number 1.

To depict correctly the many features that are unconnected from the rest of the feature network, the TIGER File design allows for the existence of 1-cell complexes unconnected from the remaining 1-cell complex. Unconnected complexes are called islands. No reason was found for linking an island to the surrounding complex through the addition of a false 1-cell. The TIGER File design also allows a single file to contain more than one primary complex, each surrounded by the unbounded 2-cell 1.

Typically a 1-cell separates two different 2-cells and such a 1-cell is called a boundary 1-cell. A 1-cell that is inside of a 2-cell is called an internal 1-cell and is bounded on both its left and right sides by the same 2-cell.

TOPOLOGICAL RULES

The preceding definitions are not sufficient to characterize the topological and geometric structure contained in the TIGER File. The TIGER File topology is more than simple sets of topological cells; it has rules that organize the sets into a unified consistent structure.

The files are subject to two basic rules. The rule of topological completeness requires that the topological relationships between cells are complete. The rule of topological-geometric consistency requires a consistent relationship between the geometric or coordinate placement of cells and the pure topological relationships of cells.

Topological Completeness

From the cell definitions it is known that 1-cells are related to Ø-cells since every 1-cell is bounded by its from and to Ø-cells and that 1-cells are related to 2-cells since every 1-cell is bounded by its left and right 2-cells. The boundary relationships in a TIGER File are its set of topological incidences. To ensure that the set of topological incidences is complete, a TIGER File must obey the following two rules:

1) The sides of 1-cells incident to a Ø-cell form a cycle.
2) The endpoints of 1-cells bounding a 2-cell form one or more disjoint cycles.

Topological-geometric Consistency

The rule of topological and geometric consistency states that the collection of topological cells must have coordinates that make the collection a disjoint partitioning of the coordinate space. The consistency rule may be decomposed into four conditions that must each be true of an entire TIGER File. The conditions are:

1) No two members of the combined set of Ø-cells and curvature points share the same coordinate.
2) No two vector interiors share a common coordinate.
3) No two 2-cell interiors share a common coordinate.
4) One cycle of a 2-cell's bounding 1-cells has an signed area greater than zero and surrounds the other cycles that have signed areas less than zero. The exception to this condition is 2-cell 1 that has one or more cycles each with an signed area less than zero.

IMPLICATIONS FOR SOFTWARE

At any given time, a TIGER File's topology necessarily conforms to the topological definitions and rules. Therefore all topological changes applied to a TIGER File must result in a file that still is valid topologically. The need to maintain topological completeness and topological-geometric consistency dictates the modus operandi of the software that manages topological changes.

The topology management software can be constructed as a library of functions that operate on cells. These are the functions normally found in automated systems such as add, delete, find, and move. Since the objects of the functions are topological cells or even parts of cells, examples of specific functions performed on a TIGER File would be add 1-cell, delete Ø-cell, move curvature point. The functions would be combined to form complete topological actions. For instance, the complete action of deleting a 1-cell requires as sub-functions the merge 2-cell function to fuse the two 2-cells that a boundary 1-cell separates and the delete Ø-cell function to eliminate a Ø-cell that no longer has incident 1-cells.

The maintenance of TIGER File topology must be through software alone. There must be no human component or function in the system that maintains the topological structure. The only function of a TIGER File user is that of retrieving and modifying the feature network, not the labelling of 1-cell sides or some such operation.

Linear Feature Insertion

The maintenance of topological consistency by software alone may be illustrated through a high-level description of an algorithm that inserts a new linear feature into a TIGER File. The algorithm's domain is the transformation of a chain of coordinate points into one or more new 1-cells resident in a TIGER File. It is not concerned with the digitization of the feature or with the attachment of attribute data after insertion.

The chain of coordinates is first edited to remove or modify illegal and undesirable portions of the chain. The edit must at least detect and resolve self-intersections within the chain. Self-intersections are resolved by the addition of points at positions in the chain where intersections exist. The new points must be marked as intersection points. Degenerate vectors must be eliminated. Optional modules of the edit are a thinning algorithm to reduce the number of superfluous points and algorithms to eliminate small knots and 'z' shapes. After completion of the chain editing, the original chain has been decomposed into substrings each of which is atomic relative to the entire chain.

The next module in linear feature insertion is the detection of all intersections between the new atomic chains of coordinates and 1-cells in the TIGER File. Where the new feature intersects the interior of a 1-cell, the 1-cell must be split into two 1-cells both ending at the intersection point. Likewise, where a 1-cell or Ø-cell intersects the interior of a chain, the chain must be split by adding the intersection point to the chain and marking it as an intersection point. Once the detection and resolution of intersections is complete, both the TIGER

File and the new linear feature reflect the intersections through the splitting of 1-cells and the addition of intersection points.

A tolerance distance should be used in the determination of intersections in addition to strictly solving simultaneous linear equations. The tolerance enables the snapping of the new feature to nearby Ø-cells and 1-cells.

The remaining module is the insertion of each atomic chain into the appropriate 2-cell. The atomic chain insertion requires an analysis of the 2-cell's 1-cell cycles in regards to the incidence of the new chain. A 2-cell labelled A is split by an atomic chain into two 2-cells, A and B, if and only if the chain connects Ø-cells on the same cycle or if the chain is a loop. When the chain is a loop the outward facing side is linked to the cycle of 2-cell A it is connected to or is added as a new cycle of 2-cell A if it is an unconnected island. When the new chain connects two Ø-cells on the same cycle, the cycle of 1-cell sides is correspondingly split at the two Ø-cells. Half of the cycle is attached to 2-cell A along with the consistent side of the new chain. The other half of the split 2-cell's cycle is completed by the other side of the new chain and is assigned to 2-cell B. To complete the 2-cell split, islands must be assigned to one or the other 2-cells. Using a point-in-polygon routine each of the cycles not split by the new chain must be determined to be inside 2-cell A or B. The point-in-polygon routine should also be used to assign point landmarks to the appropriate 2-cell. Since the child 2-cells only occupy area occupied by the parent 2-cell, they are linked to the different geographic areas and area landmarks that were associated with the original 2-cell.

REFERENCES

Corbett, James P., 1979, Topological Principles in Cartography, Technical Paper 48. Washington D.C.: Bureau of the Census.

White, Marvin S., Jr., 1984, Technical Requirements and Standards for a Multipurpose Geographic Data System: The American Cartographer, vol. 11, no. 1, pp. 15-26.

MAINTENANCE OF GEOGRAPHIC STRUCTURE FILES AT THE BUREAU OF THE CENSUS

T. McDowell
D. Meixler
P. Rosenson
B. Davis
Geography Division
Bureau of the Census
Washington, D.C. 20233

ABSTRACT

In 1985 and 1986, the U.S. Bureau of the Census developed an interactive program embodying a system of subroutines to store, review, and modify the geography of a map. The process is based upon the grouping of underlying 2-cells according to their geographic cover. These groups of 2-cells are the minimum intersections of geographic entities and designated as the Geographic Tabulation Unit Base, or GTUB. The theoretical framework for the system was developed and documented in a paper presented at Auto Carto 7 [Meixler and Saalfeld, 1985] before implementation of the system. This paper describes the adopted approach and possible extensions.

Specific routines perform the extraction of boundaries, the windowing of geographic entities, the maintenance and referencing of specified geographic entities, and the comparison of geographic hierarchies. These routines work effectively in an interactive environment and confirm the soundness of the theoretical approach. The structure upon which these routines were implemented did not allow development of the full capabilities of the GTUB routines. Various enhancements and extensions to be incorporated into these routines are discussed.

BACKGROUND

Historically, decennial census data have been collected by enumeration districts (EDs). These districts were defined to honor those geographic boundaries necessary for decennial census tabulation. The EDs were sized by estimated housing counts to provide a reasonable work assignment for an enumerator. For the 1980 census, an effort was made to reduce the geographic coding errors associated with controlling in excess of 300,000 EDs. This effort involved the advance definition and entry into the master geographic file of an explicit record containing unique combinations of geographic tabulation codes. This record was named the Geographic Tabulation Unit Base (GTUB). Later this GTUB record was used as the framework under which one or more EDs were added, depending upon the population of the area. This master geographic file contained no digital map representation, thus the GTUB was the basic representation of the geography associated with any particular area.

With the advent of the Topologically Integrated Geographic

Encoding and Referencing (TIGER) System for the 1990 census, both a specific digital map and geographic entities are integrated into a single file. This change does not lessen the number or complexity of the geographic areas required for census tabulation. The GTUB concept was analyzed and adapted to meet the requirements of this integrated structure. It was recognized as representing the atomic elements of the complete geographic lattice, thus proving useful in high level geographic comparability. As an intersection record for multiple geographic hierarchies, the GTUB reduces the pointer overhead in linking 2-cells to the geographic entities. And the multilist structure allows quick collection of the GTUB refinement of these geographic entities.

Thus the GTUB is useful in conceptually and structurally simplifying the handling of this myriad of geographic areas. As part of the TIGER File representation of multiple geographic areas, the GTUB will serve as the connection between each individual geographic entity and the land area belonging to that entity. As in 1980, each GTUB will contain a specific combination of geographic codes that represents the intersection of various geographic entities. This enables it to link together all two cells sharing this unique set of geography.

THE PROGRAM ENVIRONMENT

The program evolved out of a desire to test ideas and concepts in the routines that create and manipulate GTUBs. All work is done on the Unisys (Sperry) system using the TIGER File and a library of graphics routines. The purpose was to show graphically, using a Tektronix 4115 color terminal, geographic changes to the TIGER File.

A TIGER File is split logically into many different subfiles. A subfile is organized and can be accessed in one of two ways: as a balanced tree that is ordered and accessed by a key, or as a random access logical subfile where each record is identified by its relative position in the subfile. The GTUB records are stored in a random access logical subfile. However, each of these records has a directory record for easier program maintenance.

The screen of the Tektronix terminal is divided into three areas: dialog, instructions or information, and graphics. Graphics displayed during a session are saved on the hard disk attached to the terminal and are automatically reloaded when the file is later processed on that terminal. Since the initial display of the file takes much longer than a load of the graphics from its local disk, a batch process can be used to store all or parts of the graphics before work begins.

Initially, only the outline of the file is displayed on the screen. At this point either the entire map, individual entities, or groups of similar entity types can be displayed. Linear features are color coded according to a subset of the census feature class code. Geographic entities are displayed as colored panels behind the linear features, emphasizing their areal characteristics. Panels can be turned on and off to reduce clutter on the screen. Distinct fill patterns were chosen for each entity type. Thus geographic comparisons may be viewed as overlapping panels. The terminal has a manual zoom feature that can be used to look at areas in more detail. This interactive geographic update program allows a user to review the area of geographic entities, and to change it when necessary. In addition, the user may window on a geographic entity by entering a window command.

THE GTUB RELATED COMMANDS

The program is command oriented, with English-like commands entered from the terminal. Certain commands display a short help screen in the information area. Routines that modify geographic entities alter the GTUBs in the TIGER File directly. Modifications are then extracted from the file and displayed on the terminal for the user to review. The SHOW, WINDOW and CHANGE commands utilize the GTUB routines.

The SHOW command is used to view a geographic entity. The user chooses which geographic entity to display by specifying its geographic codes. The area covered by this geographic entity is constructed by using the GTUBs associated with the entity as building blocks. The boundary is then extracted using information from the 2-cells linked to these GTUBs and used by the panel filling routine.

For each of the GTUBs in the entity, the list linking the GTUB to its underlying group of 2-cells is examined. In this list, each GTUB points to the first 2-cell within the GTUB. Each 2-cell then points to the next 2-cell same GTUB to continue the list of area covered by this GTUB. The boundary of each 2-cell is examined using a similar list that links the 2-cell to the chain of 1-cells that define the polygon.

The boundary file for the geographic entity is created by performing a Boolean add of all 1-cells defining a given 2-cell to the boundary file; in other words, as each 1-cell is examined. If it is not found in the boundary subfile, it is added, otherwise it is deleted. This process will assure a file of distinct 1-cells with all common elements deleted. Repeated application of this procedure on all 2-cells linked to each GTUB in the entity will yield a subfile of boundary 1-cells for that geographic entity. These 1-cells are now ordered and given to the Tektronix hardware which panel fills the area on the screen. During the ordering process, the boundary cycles are determined. If more than one cycle exists, the user is informed that the entity is either discontiguous or contains holes.

The WINDOW command allows the user to fill the graphic screen with a specific geographic entity. The boundary extraction process is done as in the SHOW command. If the entity has not been displayed previously, it is then panel filled. Additionally, the minimum and maximum coordinate of the boundary are found and correctly proportioned for display purposes. They are then sent to the Tektronix terminal which adjusts the graphic display.

The CHANGE command is used to modify the areal coverage for a geographic entity. When the coverage of a given geographic entity changes, the GTUB associated with the affected 2-cell must be modified. Three subcommands have been developed to accomplish such change.

The CHANGE BY command is used in the special case where the boundary of the geographic entity to be inserted or updated encompasses the area of an existing geographic entity in the file. In many cases, only the GTUB records need be referenced. The process entails examining every GTUB associated with the existing entity. If the GTUB already has the proper codes for the new entity then this GTUB is skipped. Otherwise, the appropriate fields on a copy of the GTUB are modified to reflect the requested change. The GTUB directory is searched to see if the modified GTUB already exists. If it does, all 2-cells under the modified GTUB are merged with the list of underlying 2-cells for the existing GTUB. The old GTUB is then deleted from the GTUB directory and the random access subfile. If the modified GTUB does not exist, then the proper fields can be changed on the existing GTUB and a revised directory record created.

The CHANGE AREA command is used to insert a new geographic entity. This command allows the user to delineate an area by chaining the 1-cell boundary around it. Once the boundary is specified, the program retrieves all 2-cells within the area. The GTUB cover for each 2-cell is then adjusted. To change the geography of a given 2-cell, a copy of the current GTUB above that 2-cell is obtained. The 2-cell is then delinked from the list of 2-cell for this GTUB. If the 2-cell is the only remaining 2-cell under this GTUB then, the GTUB and its directory record may be deleted. The copy of the GTUB is then modified to have the requested change and the GTUB directory searched to see if the modified GTUB already exists. If it exists, then the 2-cell is added to the list of underlying 2-cells of this GTUB. If the modified GTUB does not exist then a new GTUB is created and added to the GTUB subfile and directory. Finally the 2-cell is linked to this new GTUB. For efficiency, the current procedure actually handles every affected 2-cell under a single GTUB in one pass.

The CHANGE BORDER command is used when a boundary update involves annexing a neighboring area. In this case, the user selects a starting point on the existing boundary. From this point, the user chains around the area to be annexed until returning to the existing boundary. All 2-cells contained in this area are determined and then changed using the method described above.

ENHANCEMENTS AND EXTENSIONS

Programs that access or alter the stored relationships between different records in the file do so through standard list handling routines that are part of the I/O management system. The list routines enable programs to add, delete, search, or modify a member of a list. Other capabilities include merging lists or moving between lists that share a common record type. All routines can work with any of the lists defined in the system. These include simple one-way, directory, intertwined, and many-to-many lists. The next release will include teo-way lists and multilists.

The multilist will be used to link the entities and their GTUBs. This list is defined by a generic name and a record type. All entity records reside in the same subfile. They are distinguished by an entity type field. Many entities will be linked to a GTUB, with more than one list threading through the record. Rather than defining each list separately, the routines will determine the correct list and the corresponding pointers by the type of entity that is being manipulated. This will free the programmer from having to be aware of many named lists.

Boundary maintenance has been of questionable efficiency during the first implementation of these routines. It has been proposed that boundaries only be maintained for the GTUBs and that they be extracted for the geographic entities. The GTUB routines can handle the Boolean add of 1-cells to the GTUB boundary subfiles during routine maintenance. Interestingly, the Boolean add should be done for both the GTUB from which the 2-cell is moving and the GTUB to which it is moving. As GTUB boundaries can be determined by the Boolean addition of its component 2-cells, so may the boundary of any geographic entity be determined from its component GTUBs. A technique that is being explored is to store the boundary segments for a GTUB as bit flags in a separate subfile for each GTUB. The GTUB routines would then be optionally responsible for maintaining these boundary subfiles. Most boundary subfiles would only comprise one physical page of data in the system. The boundary for an entity could then be extracted by gathering all the GTUBs contained in that entity and performing an XOR operation on the boundary subfiles. An integral part of the TIGER System is the I/O management system that handles the creation and manipulation of these subfiles. The routines are general purpose and assume nothing concerning type, internal name or number of subfiles within the file. This feature enables an applications program to create and use additional subfiles to store information during processing. Thus, GTUB boundary subfiles may be added easily as part of an adjunct file.

Manual comparability is implemented currently by the graphic overlay of panels. Similarly, coextension and nesting of geographic entities can be inserted into the system using exclusively the CHANGE BY command. In the future it will be useful to edit and test comparability and nesting of geographic entities by reviewing their GTUB relationships. Such high level comparisons need not go to the underlying 2-cells.

CONCLUSION

The GTUB routines have been implemented successfully on a preliminary TIGER file structure. These routines concentrate on the automatic maintenance of the GTUB records. The user appears to change geographic codes for graphically defined areas of the file. The user does not see the addition, or deletion of GTUB records, nor the moving of groups of 2-cells between GTUBs. Because of the file structure, these routines have concentrated on the GTUB maintenance and linkage between the GTUB and its composite 2-cells. The coordination of the GTUBs under the geographic entities will be an integral part of the TIGER File structure for the 199Ø census. The list handling capabilities of the I/O management system will provide the maintenance of these GTUBs under the geographic entities.

REFERENCES

Meixler,D. and Saalfeld,A.,1985, "Storing, Retrieving and Maintaining Information on Geographic Structures: A Geographic Tabulation Unit Base (GTUB) Approach", AUTO CARTO 7 PROCEEDINGS, pp 369-376,Washington, D.C.

ESTIMATING PRODUCTION PARAMETERS FOR LARGE VOLUME AUTOMATED MAPPING ON A RASTER PLOTTER

Thomas L. Wanie
Geography Division
U.S. Bureau of the Census
Washington, D.C. 20233

ABSTRACT

The Bureau of the Census will be producing hundreds of thousands of individual maps in support of the 1990 Census of Population and Housing. The map production system will be fully automated, therefore, great dependence is placed on new technology and operating procedures. Vendor specifications on equipment capabilities alone can neither provide a true measure of how much equipment and staff are required to meet the production schedule, nor determine which procedures to follow in order to maximize production flow and minimize administrative control. Since an automated cartographic production environment of this magnitude was not available from which to extract production parameters, it was necessary to develop a means of estimating them. This paper describes the content and results of the 1986 Electrostatic Plotter Production Test conducted by the Census Bureau.

Although production demands such as those in support of a census operation are somewhat unique, the results of the test reveal numerous general parameters about map production using an electrostatic plotter and provide a benchmark from which to measure performance.

NEED FOR THE CARTOGRAPHIC PRODUCT

The nature of field work requires (1) the production of multiple copies of a very large number of individual maps at many different scales, (2) the correction of features and boundaries in the data base from which those maps were made within a relatively short time frame, and (3) the generation and distribution of updated versions of those maps. This presents a formidable challenge considering the small amount of production time and the great number of maps to be produced. There will be approximately 400,000 enumerator assignment areas defined for the 1990 Census of Population and Housing. For field collection activities, each enumeration area is represented on a separate map and all maps must be delivered to their respective locations within weeks of each other. Public Law 94-171 requires that the Census Bureau provide population counts together with appropriate maps to each state for the purpose of redistricting in a matter of months after the 1990 Census is taken. Maps also are provided to state and local governments for use in reviewing the completeness of the count in operations scheduled before and after census day. Given the volume of maps to be produced, timeliness of production and delivery become the greatest consideration in the design of a map production system.

SELECTION OF EQUIPMENT

The least efficient functions of an automated cartographic system are the input and output of data. Census Bureau operations require an output device that will convert a stored digital cartographic image into a printed map sheet as quickly and inexpensively as possible. These requirements immediately exclude some categories of plotting devices. Pen plotters, while providing a range of options from monochrome to multiple colors and a variable quality of line work, are far too slow. Large format laser plotters have great potential for use but their present formats are restrictive. In addition, large format laser plotters generally create images on photographic film. To create a usable map product, the image must be transferred photographically to paper. This extra step requires unacceptably large amounts of additional time and expense. Large raster format electrostatic plotters possess the attributes best suited for production needs.

Formerly strictly a monochrome device with coarse resolution, electrostatic plotters now have color capability and increasing resolution. They produce maps that can be used in the field with no further processing other than trimming the sheet. When switching from production of one map to production of another, there is a minimum of changeover time. Often the changeover from production of one map to the next is only a matter of reading the next file from a tape, and rasterizing the map image. Most importantly, raster plotters are orders of magnitude faster than pen plotters and the operations involved in processing maps photographically. The sacrifice is in map design options. Maps had to be designed to fit the limitations of the plotter in use at the time: a monochromatic electrostatic plotter with 200 dots-per-inch resolution and an image area limit of 35.25 inches in one direction.

PRODUCTION AND PLANNING

In the early stages of planning the production and distribution of computer-generated cartographic products for field work there were a number of unknown factors. Basic parameters were difficult to estimate since there was no precedent operation. The most imperative question was the overall rate of production: how many maps could be produced during each shift?

Rate of production affects planning in many area of geographic support for the 1990 decennial census. Most directly affected by production rate are the number of plotters to be purchased, the amount of staff to be hired, and the scheduling of resources. In order for planning and purchasing to progress on schedule, valid estimates of production capacity are needed.

The rate of map production is a combination of three factors: equipment, procedures, and map design. To determine a valid estimate of production rate, both equipment and procedures require testing and evaluation. The testing should be done while producing maps similar to those to be produced in future operations. The equipment to be used had not been tested under full load conditions over a significant period so a simple "load" portion of the test would provide initial estimates of hardware reliability and productivity. The ratio of hardware "up-time" to "down time" would provide a reasonable estimate of hardware reliability. The rate that maps were read from tape, rasterized, and plotted would result in a measure of hardware productivity.

Rate of production is equally affected by the procedures used to control incoming requests, generation of an image, quality assurance, and delivery. Introducing new technology into map production requires redesign of the entire map production and distribution system. The ability to store maps as digital images on tape and create paper maps on demand precludes the need for warehousing large numbers of maps. Maps are created if and when there is a need, thereby eliminating wasted time and materials. The "on-demand" creation of a map requires a job control system that routes a request from the user, to the plotter, and back to the user quickly and accurately.

Map design is also factor influencing production rate. The number of lines on a map affect the time to read and rasterize the map from tape. The maps used in the test were from current census operations: the 1986 Censuses of East Central Mississippi and Central Los Angeles County, and the 1987 Census of North Central North Dakota. The design of these maps is similar to what is planned for future operations.

TEST OBJECTIVES

Accurate estimates of total production require testing of both equipment and procedures simultaneously. In order to gather all the facts needed for effective planning, a set of objectives were formulated:

- Operate an electrostatic plotter for a specified period under production conditions.
- Determine a realistic map production rate to validate assumed production rates used in planning.
- Develop and test a map request system.
- Develop and test operator, control, and quality assurance procedures.
- Provide experience to help refine staffing requirements for future production units.

TEST BACKGROUND

To produce maps for field work during the 1990 decennial census there will be twelve regional map plotting sites around the country, along with facilities at Census Bureau headquarters and the Data Preparation Division in Jeffersonville, Indiana. These fourteen sites will produce maps requested by over four hundred district offices. The electrostatic plotter production test simulated a regional plotting site in full production with the responsibility of providing maps for a number of district offices. The test was designed primarily to test the production capacity of an electrostatic plotter, and secondarily to test the control of work flow in a production setting. By simulating the work flow at a regional plot site in addition to a simple "load test" on the hardware, it was possible to gather valuable information for improving training procedures, production procedures, and quality control procedures, and for determining the space and equipment requirements of future operations. The site and staff for the test were chosen to approximate as closely as possible the conditions that would be present during an actual census operation. The site chosen was the Laguna Niguel Processing Office in Laguna Niguel, California. The Laguna Niguel Processing Office had been established to process the data collected during the 1986 Census of Central Los Angeles County. The site was selected because it best met the combined requirements of equipment, staff, and space necessary to simulate a production environment. Sufficient staff was available at the processing office to participate in the test since the processing of the 1986 Census of Central Los Angeles County was nearly completed. The majority of the staff there had very minimal knowledge of census geography, cartography, and computer operations. Using an inexperienced staff would more closely simulate a 1990 census production operation. The total staff for the plotter test consisted of fourteen people, who were divided into two groups of seven each. The first group worked the day shift (7:30 am, to 4:00 pm) and the second group worked the night shift (4:00 pm to midnight). Each shift included one supervisor, one assignment clerk, one plotter operator, two quality control technicians, one multi-purpose staff member, and one geotechnician. The multi-purpose staff member aided or substituted for the plotter operator, assignment clerk or Q.C. technician. The geotechnician was a Geography Division staff member from headquarters who performed both active and evaluative roles.

The major equipment requirement was an electrostatic plotter with a controller. This equipment was in place and operational at the Laguna Niguel Processing Office before the test. The controller read a 1600 bpi magnetic tape that was created on the Census Bureau's Sperry mainframe computers with a combination of software written in house and provided by the plotter manufacturer. The digital map stored on the magnetic tape was in a vector format and had to be rasterized. Conversion to raster format was done by the controller, vector by vector, as the tape was read. The number of files on each tape was limited by the complexity of the map images stored. In this test, the number of map images per tape ranged from three to sixty-two.. All tapes were generated at the Census Bureau's headquarters in Suitland, Maryland and shipped to the Laguna Niguel Processing Office along with lists of files contained on the tapes.

The procedures were written to function both in the test situation and eventually in an actual production environment in the regional plotting sites. Because there were several ways map requests were completed, a special management system was designed to maintain up-to-date information about all requests in the system. The management system comprised a set of control logs held by the assignment clerk. The control logs were organized by district office, one for each district office requesting maps. The control logs were used to record the progress and disposition of each map requested. The request form itself became a transmittal slip with information continually added to the form as it progressed through the process. At certain points in the process the assignment clerk made a photocopy of the request form and inserted it into the appropriate control log, replacing the previous version. Control log updates were made following the plotting of maps, assurance of quality, application of correction procedures, and disposition of the request. Using this method, the control logs provided an audit trail of every request passing through the system.

The most difficult procedures to write were those for the quality control technicians. These technicians were temporary employees for whom judging cartographic products was, for the most part, a new experience. In written procedures and in training, the emphasis was on usability of the map. If the map had a small error, but was still usable in the technicians' opinion, the map was accepted. If there were one or more errors that made the map unusable, it was rejected. If a map was rejected, the quality control technicians were to make a basic judgement concerning the source of the error in order to aid in correcting the map. Maps with errors made at the plotting stage could be replotted almost immediately; however, maps with errors introduced at the tape generation stage had to be recreated. In the latter case, a request was sent to headquarters for a new tape to be made and sent to the plot site. The tape replacement process was simulated by the geotechnician supplying a replacement tape after an appropriate delay.

TEST EXECUTION

The test spanned ten working days, each comprising two shifts, for a total of one-hundred sixty hours of potential production time. Training for the test staff took place on a Friday afternoon and the test began on the following Monday morning. Requests for maps were generated by the geotechnician who supplied them to the assignment clerk on a flow basis, simulating incoming orders from several district offices. After receiving a map request, the assignment clerk determined the location of the map by tape number and file position. The information was recorded on the request form and the form delivered to the plotter operator. The plotter operator plotted the maps as they appeared on the map request form by mounting the appropriate tape on the controller, advancing the tape to the proper file, and plotting the requested number of copies.

For purposes of the test, some files with known errors had been included without the operators' knowledge. If a tape could not be read by the controller, a request for a new tape was sent to headquarters. The plotter operator recorded the time required to read the tape file and plot the map. Time also was recorded for non-production time such as start-up and shut-down, routine maintenance, problem resolution, and staff breaks. When all the maps on a request were plotted, the operator removed the roll of maps from the plotter, attached the request form, and delivered the roll to the assignment clerk. The assignment clerk updated the control log to show the maps were plotted and transmitted the roll of maps to the quality control technicians. Once in the quality control area the technicians unrolled the maps, cut them into individual sheets, and checked each sheet for image quality and for the presence of major content errors. If all maps were accepted, they were packed and placed in a bin for shipping to the appropriate district office. If maps were rejected during the quality check, the quality control technician made a judgement as to whether the problems were introduced at the plotting stage or the tape generation stage. The control log was updated with results of the quality check. The quality control technicians also recorded starting and ending times for processing each map.

RESULTS

Table one shows the numeric results of the test. The overall rate of production was 80 maps per eight hour shift. It is difficult to make any estimates of a time unit lower than the shift level because of the relationship between maps plotted, sheets plotted, and tape reads. A map may be divided into multiple sheets. In this test, the number of sheets per map ranged between one and six. Each map was created after the controller performed a tape read. With a single tape read the controller read in the entire digital map from the tape, rasterized it, and plotted it on one or more sheets. If more than one copy of a map was needed, the duplicate copies did not require the tape to be read again because the information was already in the controller memory. For example: assume that the plotter operator received a request for four copies of a particular map and the map was divided into six sheets. The tape would have been mounted, and the digital map read into the controller and rasterized, taking approximately ten minutes. When all the information had been read from the tape, the plotter would have created the first map by plotting the six individual sheets. The plotting taking approximately two minutes. The plotter operator would have then programed the controller via a small keyboard to plot three additional copies of the map. Eighteen sheets (three copies of a six-sheet map) would have been plotted taking six minutes. Total elasped time would have been eighteen minutes, resulting in four maps and twenty-four sheets plotted. The most time consuming step is the tape read, therefore, the rate of production is dependent on how many copies of each map are plotted per tape read.

TABLE 1

	District Office			
	Mississippi	North Dakota	Los Angeles	Total
Number of maps plotted	373	894	340	1607
Number of sheets plotted	373	3570	340	4283
Amount of time for tape reading, rasterizing, and plotting (in minutes)	1899	4125	1480	7504
Quality control (minutes)	292	2107	250	2649
Total number of tape reads	196	293	168	657

Average maps per shift:	80	maps
Average sheets per shift:	214	sheets
Average Q.C. time per shift	132	minutes

The Mississippi maps had the greatest density of linework; the North Dakota maps had the largest number of multi-sheet maps and the Los

Angeles maps the smallest. The data show that the number of sheets used to display a map is a lesser factor affecting the reading and plotting time than is the number of features displayed on the map, since the Mississippi maps took more time to plot (per map) than the North Dakota maps.

Approximately 75% of the total test period was in production time; the remainder was non-production time, which consisted of meal breaks, staff meetings, routine plotter maintenance, and problem resolution.

Routine plotter maintenance included changing paper, adding toner & replenisher, and cleaning the rollers inside the plotter. During the test, the paper was changed 38 times with an average of 10.5 minutes per change for a total of 399 minutes of non-production time. Not every paper change was a result of the plotter running out of paper. A small number of rolls were damaged or creased and were changed before they were completely used.

During the entire test, 45 hours were spent in quality control. This is about a 1:3.5 ratio between quality control time and plotting time. This ratio shows that one quality control clerk should be able to quality check the output from three plotters.

RECOMMENDATIONS FOR FUTURE OPERATIONS

In an actual production setting, the rate of production is expected to increase as the staff gains familiarity and as improvements are made in equipment, procedures, and map design. Recommendations for future operations are for the most part based on problems encountered during the execution of the test. The majority of difficulties were with the hardware. A significant increase in production can only be achieved by obtaining hardware with some additional features. Features suggested by the plotter test staff along with the evaluation team from headquarters have been incorporated in plans to purchase plotters for production operations. The Bureau of the Census has issued a Request for Information on purchasing plotters built with desirable options not currently available.

Possibly the most important feature needed is the ability of the controller to read a raster format tape. The controller currently in use reads only vector format files and rasterizes the single map file being plotted. The raster image is lost as soon as the next file is read. The ability to save the raster image on tape and load it back into the the controller would save the rasterization time on subsequent requests for the map. Direct reading of a raster image also would allow cartographers to rasterize the map image using a mainframe computer, and use run-length encoding to write the image to a tape. This process would be far more efficient than rasterizing at plot time.

Other suggested features are the inclusion of a header record preceding each map file on the tape, and a programmable controller. A header record would contain the geographic identifier of the map such as MINNESOTA or TRACT 101. The operator would be able to enter the geographic identifier directly into the controller rather than be required to enter a relative position of the file on the tape. The assignment clerk would no longer have to determine and record the position of the file on the tape, saving time and eliminating a potential source of errors. The present controller allows commands to be entered only after the preceding command has been completed. The Census Bureau has suggested that the controller accept up to ten instructions at once and execute them in sequence. This would allow a single plotter operator to operate several plotters at once.

The possibility of using color plotters is also being investigated. Large format electrostatic plotters have been developed with full color capability. These plotters have slightly faster rasterization time because of improvements in processor design; however, the actual plotting time is increased. Most color electrostatic plotters use a multi-pass method. The paper passes over the charging head through the first toner for the first color, the paper then is rewound into the machine for a second pass over the charging head and through the second toner. This process is repeated for colors three and four. Plotting a two-color map results in the map making two passes through the machine, and therefore takes approximately twice as long as a single color plot. A full four-color plot takes approximately four times as long as a monochromatic plot. Because of the time constraints associated with the production of maps for field use, many maps would still be plotted in a single color, with multiple colors used only for special use or very complex maps.

CONCLUSION

All the objectives of the electrostatic plotter production test were achieved. An electrostatic plotter was operated for two shifts each day for a two-week period. This time span was sufficient to simulate production conditions. Over the two-week period, an average map production rate of 80 maps per shift was achieved. A 100 map-per-shift rate is realistic with improved equipment and procedures. A map request system was developed and used. Including a number of modifications, the system is usable for future operations. Plotter operator, control, and quality assurance procedures were developed. Recommendations from the test staff helped refine these procedures. Experience with staffing levels was obtained and have resulted in staffing recommendations. Recommendations were made for plotting sites with a number of different configurations.

The 1986 Electrostatic Plotter Production Test provided a basis from which to make (1) a reasonable estimate of the production rate of a large format electrostatic plotter, and (2) recommendations for future procedures, equipment, and staff. Although the rate of map production will change with the introduction of new demands on the system, improved equipment, and changes in procedures and staff, the results of the test have provided a reference point from which system evaluation may be judged.

DESIRABLE CHARACTERISTICS OF A SPATIAL DATA BASE MANAGEMENT SYSTEM

Stephen C. Guptill
U.S. Geological Survey
521 National Center
Reston, Virginia 22092

ABSTRACT

Just as rules have been developed to measure how well a data base management system conforms to the relational model, the desirable characteristics of a spatial data base management system can be specified. Spatial data base management systems must meet the requirements of conventional data base management systems as well as provide special facilities to handle spatial data. Characteristics such as the independent handling of feature, attribute, topology, and coordinate data and the support for alternative geometric representations are desired. This set of characteristics serves not only as criteria for evaluating existing systems, but also as input for future system design.

INTRODUCTION

The U.S. Geological Survey has collected, analyzed, and disseminated geologic, hydrologic, and cartographic information of the Nation for over 100 years. While most of this information has been provided in map and tabular form, an increasing amount of information is now being collected and stored in digital data sets that can be accessed and manipulated by computers. Although surveys, imagery, and maps form the basis of cartography, other spatial data can add useful information to the data base. Street addresses and geographic names are examples of other data used to refer to spatial entities. Items that are mappable, but not commonly shown on conventional maps (for example, geologic drill holes), and the information they convey need to be accommodated in a spatial data base model of geographic reality. Such data models typically consist of four major components: geometric descriptions of the spatial entities (objects), descriptive attributes for the objects, the topological relationships between objects, and feature relationships describing which objects comprise features.

Once spatial phenomena have been accurately and appropriately structured for a given model, the challenge remains for the user to extract and manipulate data and to display the resulting information in a manner relevant to a specific problem or application. Data may need to be extracted based on combinations of its geometry, topology, and attribute fields. Specifically, a spatial data base should be designed to meet the requirements of users who wish to define application-dependent sets of information, and spatial data base management systems need to provide the necessary functionality to meet these requirements.

Publication authorized by the Director, U.S. Geological Survey. Any use of trade names and trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the U.S. Geological Survey.

THE DATA BASE ADVANTAGE

Most of the systems that have been developed to perform an automated cartographic or a geographic information system (GIS) type of processing have not utilized data base management system (DBMS) facilities. Viewed in the abstract, DBMS technology provides significant desirable characteristics for handling spatial data. These advantages include capabilities to store interrelated data without harmful or unnecessary redundancy to serve one or more applications in a optimal fashion; to make applications programs independent of the underlying data; and to provide a common and controlled approach to add, modify, and retrieve data from the data base (Martin, 1975). The capabilities offered by relational DBMS's to respond to ad hoc queries would seem to make them ideal for handling spatial data. However, only recently has DBMS software been used to handle spatial data (for example, Morehouse, 1985; Palimaka and others, 1986). Why has it taken over a decade to develop systems with some spatial data base management capability? Is the data base advantage no advantage at all when it comes to handling spatial data? The contention here is that the DBMS environment is fundamentally the preferred environment and that the special requirements for handling spatial data have delayed the development of such systems.

SPATIAL DATA PROCESSING ENVIRONMENTS

Systems that perform automated cartography or GIS types of functions need to handle large amounts of geometric data. This fact alone sets the spatial data processing environment apart from other DBMS environments. The high-level functions performed for automated cartography or GIS's are identical (Guptill, 1985). However, the operating environments in which those functions are applied are different for automated cartography and GIS, and thus impose different requirements on a spatial DBMS.

Automated cartography activities are now being performed on stand-alone workstations with independent computer processors, memory, and disk storage. Activities such as data collection, editing, symbolization, and display are performed on a data file that represents the spatial information for a given geographic area (for example, a 1:24,000-scale map sheet). The work is performed as an independent operation; data being processed is not accessed by others. Rapid system response time is a key to efficient production operations. Thus, efficient access to the spatial data becomes a key requirement for a spatial DBMS. However, inefficient implementations of spatial access mechanisms have caused some vendors to continue to forgo utilizing DBMS technology in favor of internal memory and file management operations. This solution does not help to satisfy the additional DBMS requirements imposed by GIS's.

GIS operations may take advantage of the same types of hardware as automated cartography systems, but a key difference in operations is the need for GIS's to support multiple-user access to a central spatial data base. Features such as security, integrity, synchronization, and recovery take on added importance in this environment. Support of multiple versions of a set of relations may also be required.

EXTENSIONS REQUIRED FOR SPATIAL DATA HANDLING

A major key to creating a spatial DBMS is support of spatial data types as an integral part of the DBMS. Extensions of the relational model to support points, lines, and polygons should overcome a number of the problems that exist in trying to use general-purpose data base systems today. Some researchers

(Waugh, 1986) have worked around this problem by storing coordinate data in various data types that allow for the bulk storage of up to 64,000 bytes (such as the LONG data type in ORACLE or the SEGMENTED STRING data type in VAX Rdb/VMS).

While these solutions allow for the integrated management of all the data, they fall short in the second major area, the extensions to the model to add a set of operators for these data types. One set of such operators has been defined (Claire and Guptill, 1982) and other sets are possible. Clearly this problem area is not limited to cartography and GIS, but is germain to various engineering applications such as CAD/CAM. Some of the required extended semantics for the relational model are described in detail by Stonebraker 1986.

The third major extension concerns the expression of spatial queries. Questions such as "find all the schools in Fairfax County that are further than 5 miles from a fire station" would be difficult to express in a query language such as SQL. Presumably the addition of spatial operators would eliminate some of these difficulties by adding such terms as "spatial intersection" to the SQL language. The handling of spatial queries is an area that would benefit from some type of natural language query parser. The addition of query capability using graphic displays and a mouse also would be of assistance to cartography and GIS users.

PERFORMANCE CONSIDERATIONS

The previous section has mentioned some of the basic extensions that could be made to relational DBMS's to allow them to handle spatial data. However, even if these extensions are implemented, it does not necessarily mean that the system will have an acceptable level of performance. Assuming that the spatial operators are implemented efficiently, there are two other areas that would appear to be key to system performance: query optimization and spatial search mechanisms.

The purpose of a query optimizer is to choose an efficient strategy for evaluating a given relational expression. The optimization process has been described in terms of four stages: (1) cast the query into some internal representation; (2) convert to canonical form; (3) choose candidate low-level procedures; and (4) generate query plans and select the cheapest (Date, 1986). Each of these stages will be complicated by the addition of spatial data types and spatial operators. For example, if spatial operators are treated as a series of restrictions on a given relation, an optimizing strategy that executes all restrictions before executing joins may be inappropriate. A more appropriate sequence may be relational restrictions, relational joins, and then spatial restrictions. Optimization routines also utilize information about the current state of the data base, including the existence of indexes. Acceptable performance for spatial data handling will probably hinge on an efficient spatial search and indexing technique.

A number of researchers have been investigating various one-dimensional access structures for use with spatial data. These include QUAD trees, K-D trees, grid files, Peano keys, R-trees, and K-D-B trees (see Kleiner and Brassel, 1986; Guttman and Stonebraker, 1983). Implementation and test results for the various methods are not generally available although some success has been reported using a variation of a Peano key index within GEOVIEW (Waugh, 1986). Rapid spatial retrieval also depends on an effective physical design of the data base. Issues such as buffer management and physical clustering of coordinate data on disk may be critical in engineering an efficient system.

SUMMARY AND CONCLUSIONS

The major desirable characteristics of a spatial data base management system have been presented. Spatial data base management systems should provide the functions of existing relational data base management systems, as well as special facilities to handle spatial data. The handling of spatial data will require the definition of spatial data types and operators, design of spatial indexing, query formulation and optimization strategies, and engineering of efficient access to mass storage devices.

REFERENCES

Claire, R.W., and Guptill, S.C.,1982, Spatial Operators for Selected Data Structures: Auto-Carto V, Arlington, Va, Proceedings, August 22-28, 1982, p. 189-200.

Date, C.J., 1986, An Introduction to Database Systems: Reading, Pa., Addison-Wesley Publishing Co., p. 333-359.

Guptill, S.C., 1985, Functional Components of a Spatial Data Processor: Auto-Carto 7, Washington, D.C., Proceedings, March 11-14, 1985, p. 229-236.

Guttman, Antonin, and Stonebraker, Michael, 1983, R-Trees: A Dynamic Index Structure for Spatial Searching: Memorandum No. UCB/ERL/ M83/64, College of Engineering, University of California, Berkeley, 28 p.

Kleiner, Andreas, and Brassel, Kurt, 1986, Hierarchial Grid Structures for Static Geographic Data Bases: Auto Carto London, September 14 19, 1986, Proceedings, p. 485-496.

Martin, James, 1975, Computer Data-Base Organization: Englewood Cliffs, N.J., Prentice-Hall, Inc., p. 17.

Morehouse, Scott, 1985, ARC/INFO: A Geo-Relational Model for Spatial Information: Auto- Carto 7, Washington, D.C., March 11-14, 1985, Proceedings p. 388-397.

Palimaka, John, Halustchak, Orest, and Walker, Ward, 1986, Integration of a Spatial and Relational Database within a Geographic Information System: Technical Papers, ACSM-ASPRS Annual Convention, Vol 3., p. 131-140.

Stonebraker, Michael, 1986 ed., The INGRES Papers: Anatomy of a Relational Database System: Reading. Pa., Addison-Wesley Publishing Co., p. 313-392.

Waugh, Thomas, 1986, The GEOVIEW Design: Dept. of Geography, University of Edinburgh, 34 p.

TIGRIS:
TOPOLOGICALLY INTEGRATED GEOGRAPHIC INFORMATION SYSTEM

Dr. John R. Herring
INTERGRAPH Corporation
One Madison Industrial Park
Huntsville, Alabama 35807-2180

ABSTRACT

A Geographic Information System (GIS) requires both interactive edit/query, and powerful spatial analysis capabilities for large volumes of geographic data. This spatial analysis requirement has led to the investigation of the mathematical field of topology. The major problems of previous topological systems have been the construction of a logically consistent topological structure and the integration of structured spatial data with nonspatial attribute data. The TIGRIS system incorporates both types of data into a single, highly interactive data base using an object oriented system on Intergraph's stand alone workstations. This paper describes TIGRIS and its design.

INTRODUCTION

In this section, we describe the problems TIGRIS was designed to solve and outline the approaches taken.

The primary problem of any GIS is the manipulation and query of large quantities of spatial data. The accepted theoretical solution is to topologically structure the spatial data. TIGRIS has a spatial structure that is based on a generalization of two dimensional cellular topology (Switzer, 1975), implemented with sound mathematical principles, and provably correct algorithms.

Many systems built on topology require post processing to create the topological structure from coordinates input; separating data gathering, and data structuring functions. This is unsatisfactory for three reasons: it makes feedback from structuring impossibly slow; it limits or eliminates "what if" spatial queries; and it places the burden of defining spatial relations on the coordinates alone, which have inherent metric inaccuracies introduced by finite precision, finite representation of curves, and statistical limitations of data gathering technology. In TIGRIS, the extraction and structuring of spatial data occur in one process. Data is taken directly from the digitized input and placed into the topological structure, using algorithms optimized for topological extraction. TIGRIS also uses topology to automate and optimize analysis of the spatial data, relying heavily on the techniques of algebraic topology to limit computational loads. This makes TIGRIS a truly interactive spatial data base, that can update its topological structure "on the fly" while the user is digitizing or editing features.

A shortcoming of many GIS systems, that can cause performance problems, is the segregation of spatial and attribute data into separate management systems. This gives the system two different data interfaces, doubling the complexity of I/O processing. TIGRIS combines spatial and attribute data into a single data base, supported by an integrated query language and software interface. This is made possible by basing TIGRIS on an object oriented data management system and programming environment (The Object Manager). Response times are maintained even with large data volumes through several approaches; optimized algorithms, spatial indexes (r-trees) to organize and cluster the data, and a powerful dedicated processor (InterPro/Act 32C, UNIX based on the Fairchild Clipper processor (5 million instructions per second), with multiple support processors). Further, TIGRIS supports complex, nonhomogeneous features (composed of arbitrary combinations of points, lines, and areas), multiple representations of features, and arbitrary feature to feature relations.

The result is TIGRIS, a highly interactive, powerful, spatial and geographic information system.

THE OBJECT MANAGER

The TIGRIS software environment is based on a design methodology called "object oriented programming". The Intergraph Object Manager (OM) combines data base management, I/O control, and process control into a complete object oriented design and implementation environment.

Object oriented programming begins with the "encapsulation" of data into structures based on usage. All data structures of one type are called a "class"; each occurrence of a class is an "object" and an "instance" of its class. Each class is associated to a set of procedures ("methods") for the creation, edit, query, and deletion of instances of the class. The methods define the interface to the instances of a class, ensuring a high degree of modularity through "data hiding" (only a class method can refer to the structure of a class). Each method is invoked on a particular object by a "message" send. Classes share "message protocols" through methods with identical names and parameters. Classes can be defined using other classes as part of their structure through "subclassing", and can "inherit" all methods and messages of the "parent" class. Subclassing and inheritance increase the reusability and modularity of the methods.

An object structure may include any number of named "channels" which hold pointers to other objects. Objects are related to one another through matched pointers in channels. Channel specifications determine whether the objects on a channel are ordered, and whether the relation is one to one, one to many, or many to many. Channels can be used to group objects for collective messages.

The abstract collection consisting of the class specification, methods, and messages is referred to as a "package". A package is a fundamental grouping in object oriented data abstraction.

OM takes care of creating and deleting objects and channel connections. OM further controls message sends, including any needed indexing and I/O. Thus, OM is a data base management system as well as a object oriented software environment.

More complete descriptions of the object oriented software philosophy, methodology, and benefits can be found in the references (Cox, 1986, and White, 1986).

THE TOPOLOGICAL STRUCTURE

The concepts that the TIGRIS system uses to organize and manipulate spatial data are derived from cellular and algebraic theories of topology. Topology is "coordinate free geometry"; the study of those geometric concepts that can be defined independent of any coordinate system. A concept is coordinate system independent if a continuous change in the coordinate system does not alter its definition. For example, a curve can be defined as a continuous image of an interval of real numbers. Changes in the coordinate system modify the defining function from the interval into the space, while the curve, as a ordered point set, remains the same. Since "curve" can be equivalently defined in all acceptable coordinate systems, "curve" is a topological concept. Other such topological concepts include connected, adjacent, dimension, bounded, boundary, inside, outside, and orientation.

Why is topology useful in spatial data systems? Without coordinate geometry, topologists are forced to frame geometric theory in symbolic terms, to translate each geometric problem into an equivalent symbolic (algebraic) problem, to solve the symbolic problem, and to retranslate the results back into the geometric world. In this translation, the topologists gained a powerful symbolic tool for manipulating facts about geometric configurations. It is upon this tool that TIGRIS is based.

In using topological concepts to describe geographic objects, we split the spatial objects and relations away from their coordinate descriptions. These objects can then be manipulated without reference to their coordinate descriptions. Since the manipulation of coordinate descriptions is the usual bottle neck in spatial analysis, the coordinate-free, topological algorithms are often significantly more efficient than the coordinate based algorithms that they replace.

Topological Objects

The building blocks of two dimensional topological theory consist of points (nodes, 0-cells), nonintersecting curves between these points (edges, 1-cells), and the connected two dimensional areas bounded by these curves (faces). TIGRIS faces are allowed to have "holes" or "islands", making them distinct from topological 2-cells.

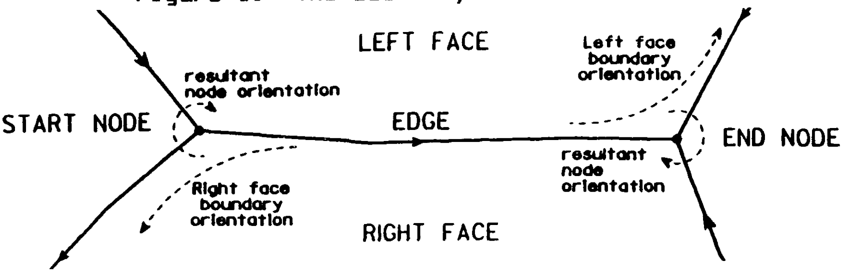
At each point in the life of a TIGRIS data set, the allowable coordinate values are divided so that each point is on one and only one topological object. At the very beginning, after the set has been initialized, but before any digitization occurs, the data set consists of a single face, that covers the entire coordinate space (called face_1 in TIGRIS). During the addition of spatial data, one of several things may occur: an isolated node may be added to the interior of some face; a node may be added to the interior of some edge, splitting that edge into two edges; an edge may be placed between two nodes in such a manner as to not split a face; or an edge may be placed between two nodes splitting a face into two faces.

Topological Relationships

The topological structure of a map resembles a jigsaw puzzle. The edges and nodes are analogous to the cut lines of the puzzle, and the faces to the puzzle pieces. The difference is that in the topological structure, the "pieces" are aware of their relationships with other adjacent "pieces". This means that the topological puzzle can assemble itself, since each piece "knows" how it is related to its surroundings and, thus, how it is related to the entire puzzle.

The relations between the topological objects are based on the notion of a "boundary". The boundary of a node is empty; the boundary of an edge consists of the two end point nodes of that edge; and the boundary of a face consists of the all the nodes and edges "close to" the face (including any isolated nodes). The coboundary relation is the reflection of the boundary relation. The coboundary of a face is empty. The coboundary of an edge consists of the two faces on either side of that edge. The coboundary of a node (node star, node umbrella) consists of all of the faces and edges that surround that node. The remaining topological information relates to the order in which things appear in the boundary and coboundary of the topological objects. The general term for this order is "orientation." The orientation of the topology derives from the establishment of a direction for each edge corresponding to the order in which its coordinates are stored. The end points of the edge are called the "start node" or the "end node" depending on whether the node point is first or last in the edge coordinate list. The two faces either side of the edge are called the "left face" or the "right face" depending on the sign of the angular direction of the face as locally measured from the tangent to the edge (left is positive, corresponding to a positive angle as measured from the tangent). The orientation of the boundary of a face, derived from "left is positive", is counterclockwise for the exterior boundary components and clockwise for the interior components. Thus, the boundary of the face consists of some number of circular lists of signed edges alternating with nodes. The sign of the edges corresponds to the leftness or rightness of the face with respect to the edge. Each isolated node is included as a separate boundary component. The orientation of the coboundary of a node is induced from the orientation of the surrounding faces. At first it is a bit counter-intuitive, but this orientation is in fact clockwise (rightward or negative) and not counterclockwise (see Figure 1).

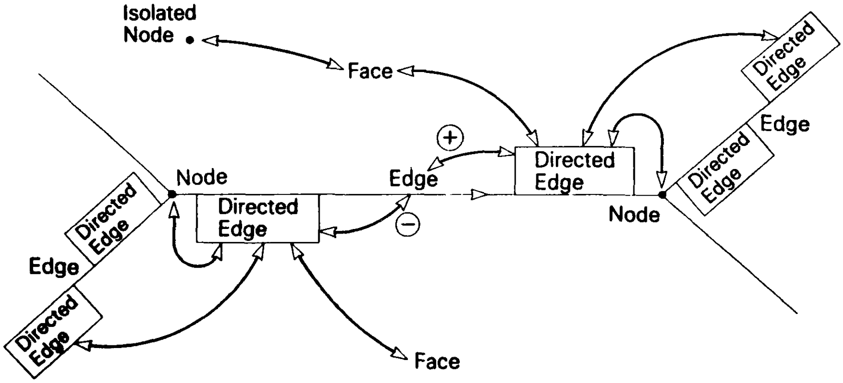
Figure 1: The Geometry of Orientation


LEFT FACE
Left face boundary orientation
resultant node orientation
START NODE
EDGE
END NODE
resultant node orientation
Right face boundary orientation
RIGHT FACE

When designing the algorithms, it becomes apparent that relationships to edges fall symmetrically into precisely two types: positive or negative orientation. To simplify code, and to eliminate orientation fields within channels, a new object type, called the "directed edge", was introduced. A directed edge is used ,to represent one of the two orientations of an edge and can act in place of the edge when a particular orientation is needed. There are precisely two directed edges for each edge: the positive and the negative incarnations.

It is important to distinguish the concept of directed edge from that of edge. An edge is a topological entity that has a spatial definition. A directed edge represents an oriented relationship with its associated edge and is not truly a geometric object at all. The directed edge can be considered as an alias for its underlying edge. To preserve the order of the directed edges about a face, links are provided from each directed edge to the one next counterclockwise about the face (see Figure 2).

Figure 2: The Geometry of the Topological Classes



THE FEATURE STRUCTURE

At the topological level of TIGRIS data, each object corresponds to a fundamental geometric entity (node, edge, or face). The next level collects topological objects into features components. Feature components are subclassed as point, line, or area depending on whether they are composed of nodes, directed edges, or faces, respectively. Feature components represent the simplest, physically homogeneous features represented in the data, (road segment, river reach, forest stand). The next and subsequent levels collect feature components and other features into more complex and abstract entities (river systems, roads, administrative districts).

These classes provide the data structures with which a schema can be built. In the schema, each cartographic entity is defined as a "dynamic class." A dynamic class definition consists of three pieces of information: the name of the base class (point, line, area feature component, or feature), a list of attributes, and any additional channel restrictions in terms of which dynamic classes each object channel may contain. For example a river system might have a "name" attribute and its composed of channel might contain stream segments and lakes (see Figure 3). All inter-object relations are in Figure 4.

Figure 3: A Complex, Heterogeneous Feature

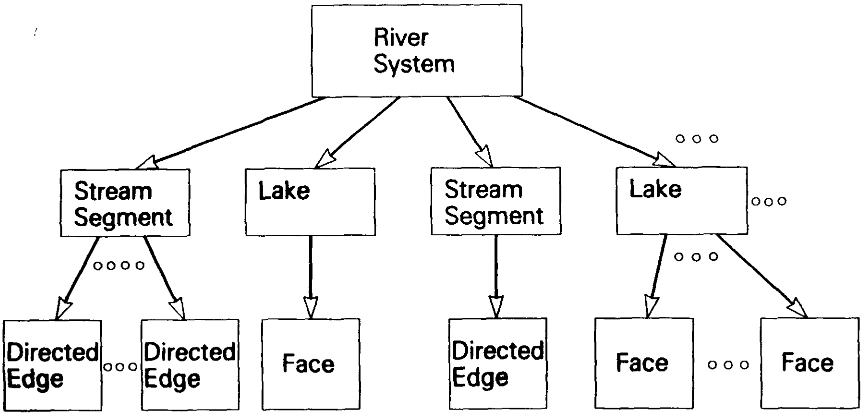
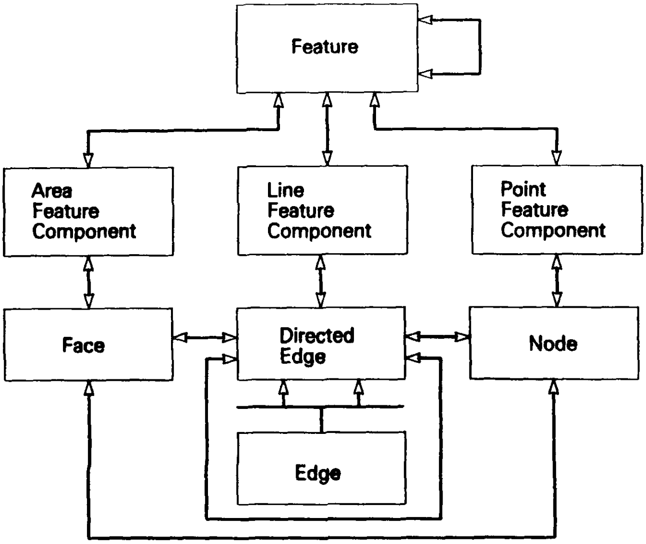


Figure 4: Class Relations



ALGORITHM DESIGN

This section describes some basic considerations in the algorithm design for the edit and query of TIGRIS data, and shows how the object oriented methodology has led to an efficient implementation of the ideas explained previously.

Spatial Localization: Intersection Search

One of the keys to the speed of the TIGRIS editor is the localization of algorithms. The topological structure expresses geometric relationships without reference to the coordinate data. Thus, if a process can be made to deal with the geometry in a localized manner, then topological linkages can be used to traverse the data in the order the process needs it. Thus, process time becomes a function of local complexity.

A example of this localization is the process of adding a new line feature to the data set. In this process, a coordinate string is specified and the topology updated to add the appropriate edges to the data set. The update of the relations for the new edge, although not simple, is straight forward, once the two associated nodes and the face to be crossed have been determined. The bottle neck is the search for the intersections of the new coordinate string with itself or with existing topology (the position of the nodes). The algorithm described here is a simplified version of the one used in the editor, but it suffices to illustrate the point.

The first step is to locate the position of the starting point of the new line feature with respect to the topology. A global r-tree index search (a subject for another paper) of the data locates the face, edge, or node upon which the first point lies. This is an index send of the query message "is on" to all topological objects, nodes, edges, and faces. OM terminates the send mechanism when a successful answer is obtained. If necessary, a node is added at this point.

With the determination (or creation) of the starting node and the specification of the second point, the add line process localizes. If the first node is isolated, the first intersection of the new line with the existing topology will occur on the boundary of the face in which this node lies. When the first node is not isolated, the exit angle of the first coordinate determines which face (or edge) the line moves through. This is a send of the query message "compare exit angle" from the node to each associated directed edge. In all cases, the search for the next intersection is a local problem.

Once the line has moved away from existing edges and nodes, for each new point, the line segment from the last point is compared with the edges and nodes bounding the face being crossed. Since the face is now known to the line method, a "find cross" message is sent to it with the new line segment. The face passes the message on to its isolated nodes and directed edges. The directed edges pass the message on to edges and nonisolated nodes. If no intersection is found for this face, then it is known that no intersection exists and the next data point is taken. Once an intersection is found, the topology is modified and the feature is integrated into the data set up to the point of intersection (now a node).

At this node, the exit angle of the line is compared with existing edges to determine which face or edge the line will follow when exiting the node. A new face is established and the process repeats until the last point in the line is placed. The entire add line process is accomplished with minimal unproductive intersection search.

Spatial Factoring: Green's Formula and Point-In-Polygon

Another key to programming in TIGRIS, is the spatial factoring of algorithms. This approach to algorithm design breaks the process into subprocesses determined by the spatial subcomponents.

A prime example of spatial factoring is the line integral used to calculate the area and orientation of polygons such as face boundaries. Green's formula (a special case of Stokes' theorem) equates twice the area enclosed within a positively oriented closed curve and the integral of x dy - y dx along the curve itself. Integrals are the prime example of spatially factorable problems. The reason for this is that if three points, P1, P2, and P3, lie on a curve, and G(.,.) is an integral expressed as an operator on the limits of integration then

$$G(P1,P2) = G(P1,P3) + G(P3,P2).$$

and

$$G(P1,P2) = - G(P2,P1)$$

Thus, Green's integral around the boundary of a face is equal to the sum of the integrals along the associated directed edges. Further, the integral along a directed edge equals the integral along the edge multiplied by the orientation. Thus, a face, queried for its area, zeros a variable and sends it as a parameter in a Green's integral message to all of its directed edges. The directed edges pass on the message to the edges, adding their orientation to the parameter list. The edges, which store the geometry, calculate the integral, multiply by orientation, and sum in their contribution to the passed parameter. When completed, the parameter is the area of the face.

The point-in-polygon test for faces is similar. The problem is, given a coordinate point and a face, to determine if the point is interior to the face. The classic solution is to intersect a ray from the point to infinity (such as the one parallel to the x-axis) with the boundary of the face and count the intersections (ignoring noncrossing tangents). If there are an even number of crossings (or none), then the point is not in the face. If there are an odd number of crossings, then the point is interior to the face. In TIGRIS, this has been modified to count crossings with orientation, 1 for an upward going crossing and -1 for a downward going crossing. The answer returned by a face is either 0 or 1 (-1 is possible for face 1). The crossing count function acts much as an integral, except that geometric observations can simplify the calculations. If the edge's minimum bounding rectangle does not intersect the ray, then the edge's crossing count is zero (no cross can occur). If the edge's minimum bounding rectangle does not contain the point, then the count is the same as that of the line joining the start node and end node of the edge (other crossings cancel in pairs).

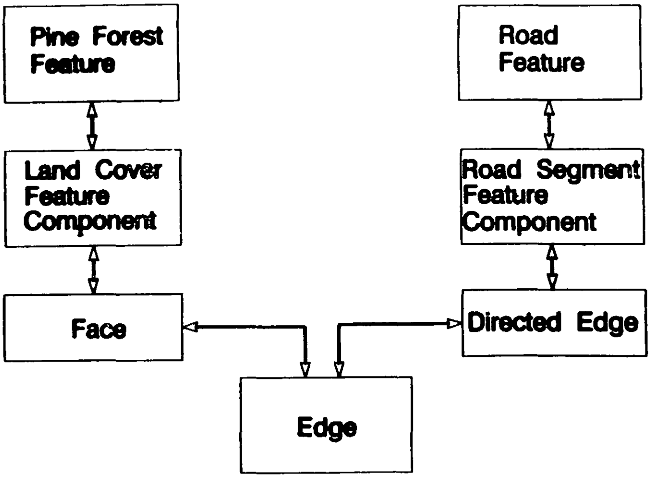
In the point-in-polygon, the factoring did not lead directly to an increase in efficiency, but allowed optimization based on geometric reasoning. In the topological version, each edge, except those whose minimum bounding rectangle includes the point in question, is treated as a whole, the count made using only the end points. In real data, the number of coordinate points in an edge can average 50 to 100 points. Thus, an increase in algorithmic efficiency by a factor of 50 to 100 can be reasonably expected.

Spatial Without Coordinates: Adjacency

In TIGRIS, all features derive their spatial extent through relationships to topological objects and all topological relationships are explicitly stated or easily derived from explicitly stated relationships. Thus, all queries based on topological relationships can be answered without reference to coordinates.

Consider the query "select all pine forests adjacent to roads." The forest half of the operation entails finding the pine forest, following a sequence of channels to locate all faces that make up the pine forests, and then following the face channels to all edges that lie adjacent to the pine forest (see Figure 5). The road half of the operation follows line to directed edge to edge channels to find all edges that are roads. By comparing these sets of edges, the query can be answered easily.

Figure 5: An Adjacency Query Path



An easy implementation of spatial query would use a query mask stored within the objects. In the example above, a message is sent by an index on forest type to the pine forest, and is passed on to the pine forest faces. The faces pass the message to the directed edges which pass the message to the edges to clear the query mask. A message is sent through roads, to directed edges, to edges to set the mask. A final message is sent through the forest (to faces, to directed edges) to the edges to check the query mask. The edges with the query mask set are the ones where roads lie adjacent to pine forest. The pine forest object sending the message is the one adjacent to the road. The sequence of messages have identified the pine forest adjacent to the road as well as the location of the road.

The time required to perform this spatial query is the sum of two message propagations through the pine forest and one propagation through the roads. This is a significant difference from the problems involved in a pure spatial intersection. Further, the algorithm takes great advantage of the parallelism of the message-send subroutine stack and the object relation graph, since when a edge finds a set query mask, it returns a success code directly to a method on the member of pine forest that is its owner.

CONCLUSIONS

The synergism between the topological representation of geographic data and the object oriented design philosophy has enabled TIGRIS to meet its major design goals of providing the geographic user with an interactive edit, query, and spatial analysis tool based upon integrated topological, spatial and attribute data. This in turn, should allow TIGRIS to redefine the capabilities expected of a GIS.

References

Artin, E., and Braun, H., 1969, Introduction to Algebraic Topology, Charles E. Merrill Publishing Company, Columbus, Ohio

Cox, B. J., 1986, Object Oriented Programming, An Evolutionary Approach, Addison-Wesley Publishing Company, Reading, Massachusetts

Lefschetz, S., 1975, Applications of Algebraic Topology, Springer-Verlag, New York

Switzer, R. M., 1975, Algebraic Topology - Homotopy and Homology, Springer-Verlag, Berlin

Spanier, E. H., 1966, Algebraic Topology, McGraw-Hill Book Company, New York

White, E., and Malloy, R., ed., 1986, Object Oriented Programming, BYTE, August 1986, pp 137-233

AN INTEGRATED DBMS APPROACH TO GEOGRAPHICAL INFORMATION SYSTEMS.

M S Bundock
Cambridge Interactive Systems Ltd.,
Harston Mill, Harston
Cambridge CB2 5NH, U.K.

ABSTRACT

The key features of data continuity, topology, and very large data volumes are discussed in relation to Geographical Information Systems. A database management system approach is suggested, and the problem of response time degradation addressed. Data analysis techniques, based on Information Engineering methodologies, have been used to produce a datamodel representing the entities and relationships within the geometric data. The datamodel, which progresses beyond the standard point-line-polygon structure, is presented and described. Its applicability to a wide range of GIS applications is demonstrated, and an implementation using relational DBMS, high level language, and data dictionary technology mentioned. The implementation uses a relational DBMS to manage both geometric and attribute data in a single spatially continuous database.

INTRODUCTION

The explosive growth in interest in Geographical Information Systems (GIS), as opposed to conventional mapping systems, has evolved predominantly by the growing awareness of information, of which only part may be displayed on a map. In addition, the following issues have been raised:

- that the information displayed on maps/charts etc. is often a subset of the information that is held or required elsewhere within an organisation,

- that the same information may be displayed on a number of cartographic products, each with different purpose, scale and symbology,

- that segmenting the area into discrete partitions, imposes extreme limitations, especially as we are attempting to model a continuous phenomenon - the land, and

- that to maximise the usefulness of the organisation's data, flexible tools that link both cartographic (graphic) and geographic (non-graphic) data should be available.

Historically, the CAD/CADCAM and the general data-processing industries have in many ways have followed separate but parallel development paths. Each industry responded to the perceived requirements of their users, while very few users recognised the need for a combined integrated approach. As a consequence the products that were available to users of geographical data who required cartographic output, tended to become little more than slightly adapted computer aided drafting systems.

By utilising the developments within both the CAD and general DP industries simultaneously, we should be able to better manage an organisations most precious asset - its data, and present it in the most suitable manner to convey information. In particular, the use of DBMS technology provides us with a method of managing the data while CAD techniques provide us with a means of efficiently presenting the data graphically.

Developments associated with database management systems, have provided us with methods for analysing data and its structure to create a "logical" database design. These methods can be applied to cartographic data in just the same way that they might be applied to any other business data. The cartographic entities and their inter-relationships, may be represented in an easily understood manner using the relational model (Date 1986).

Implementation of the model, using a relational DBMS, requires that we must address the problems of performance by considering "physical" database design. The volume of cartographic data that is displayed and interrogated, in a typical interactive GIS environment, is significantly greater than that associated with more traditional DBMS activities. Techniques that minimise disk I/O can have a significant effect on overall performance.

Throughout this paper, it is assumed that the cartographic data is being represented using vector, rather than raster techniques, unless stated otherwise. The author wishes to emphasise that this is not meant to be a statement that vector techniques are always more appropriate than raster, rather that vector representations can be used in a wide variety of applications. Clearly, raster overlays from for instance, satellite imagery and aerial photography, will often be useful in helping to convey information to a GIS user.

HISTORY

Vendors of mapping and Geographical Information Systems, create/release products in response to perceived user requirements. Until recently, the perceived user requirements have been dominated by those of the drafting/surveying/engineering departments, whose main concern was to produce their products (drawings/maps) more efficiently. As cartographic applications were seen to fall outside the realm of the DP department, the drafting departments were responsible for their own cost-benefit justification and expansion program. Throughout the 60s and early 70s the demands placed on CAD vendors were principally graphics-oriented. In the mid 70s we saw an increasing demand to allow the association of non-graphic attributes to graphic entities.

During the same period, DP departments had been developing/installing business applications (e.g. payroll, MIS, inventory etc.) which involved the management of large volumes of data, often with complex inter-relationships between data entities. This led during the 60s and 70s to the development of database management systems, with facilities for managing very large volumes of data, having complex data relationships, in a multi-user environment. Also, data processing vendors started supplying more generalised and efficient capabilities for forms/screen management, query languages and high level languages for applications development.

Once non-graphic attributes could be associated with graphic entities, many CAD equipped drafting departments began duplicating the data and data capture effort already undertaken by the users of facilities supplied by the data processing department. Most of us have seen examples of the same data being duplicated within an organisation. Where there is no coordinated effort to capture, validate and update that data, the likelyhood of inconsistency and error grows immensely.

During the late 70s and early 80s we have seen acceptance of the concept of "corporate data", i.e. the data captured by an organisation should be accessible throughout the organisation, be owned by a section of the organisation, and preferrably not duplicated. The data can then be used to convey information to users in a consistent manner, and reduce the possibility of inconsistency and error. The data may be presented to each user in a different form, but the source should be the same. In this way the user extracts the required information from the data. The representation of a pipe as a line on a graphics screen, is merely one way of conveying information about that pipe to a user. The data describing this representation of the pipe is perhaps its x,y,z coordinates. The information conveyed to the user is the whereabouts of the pipe and its relationship to surrounding objects.

PERCEIVED PROBLEM AREAS

Many existing systems available today exhibit a number of significant deficiencies from the users point of view, including:-

- Inability to manage and associate large volumes of attribute data to graphics entities, without partitioning or replicating the data,
- Inability to create and maintain complex relationships between data entities, be they graphics or non-graphical attributes,
- Inability to give a performance level that is basically independent of data volume,
- Inability to provide general multi-user access to data,
- Inability to provide adequate access security,
- Inability to provide adequate data integrity validation,
- Inability to provide a continuous mapping capability, and hence network analysis, derivative mapping etc. become very difficult,
- Inability to create products for a number of uses at a wide range of scales from a single database representation,
- Inability to model the topology often associated with graphical elements.

Many of the above problems arise from the use of special display file structures, designed specifically to allow user interaction with a logical map sheet. The map sheet (design file, coverage etc.) concept followed logically from the perceived requirement to computerise the map production cycle, i.e. produce a computer version of a piece of paper. Also it conveniently broke the total data volume down into bite-sized pieces that could be more readily handled and give adequate performance. However, our world is not segmented into discrete rectangular sections corresponding to map sheets, and there is really no reason why we should not model the world as it really is, continuous.

DBMS SOLUTION

Many of the above problems can be solved by using any one of a number of commercially available database management systems. Hence it is proposed that a DBMS be utilised to manage not only the geographic attribute data, but also the cartographic data. The data defining the cartographic entities to be displayed can be managed within a DBMS in much the same way as any other data.

Clearly the volume of data associated with providing a continuous cartographic coverage of a country/state/county is large, but is often not significantly greater than that associated with geographic attribute data for the same area. Preliminary estimates of data volumes, for a range of GIS applications (including public utilities, cadastral, topographic and DTM) providing coverage at a national or county level, indicate that total data volumes would typically vary between 1 and 100 Gigabytes per organisation.

A relational DBMS provides a natural way of storing data and modelling the complex inter-relationships between data entities that occur in the real world. Relationships between data entities may be naturally expressed by each relation (table) sharing some common attribute (field), or attributes sharing a common domain. Hence a relational DBMS is able to do away with pointers (at least at the user perception level), and consequently access paths (navigation) need not be pre-defined and restricted, as with network/hierarchic systems.

Multi-user concurrency is a standard feature implemented in virtually all commercially available DBMS. Locking mechanisms are used to ensure that each user obtains a consistent view of the database, even while another user may be performing updates. Once concurrency control is implemented in the GIS environment, multiple users can view the same geographical area, each having a consistent view of that area, and each able to update it. However, deadlocks can occur when different users lock records in a different order. Most DBMS's check for deadlocks and resolve them by backing out one of the transactions.

Security mechanisms are used to provide data privacy, at either the relation (table) level, or the attribute (field) level. Hence particular types of data can be protected from inadvertent user update or viewing. Furthermore, recovery from a failed transaction (or possibly from a "what if" scenario) is possible, since DBMS's provide a facility for reversing out the incomplete changes. This is called rollback or backout. Recovery from a corrupted database is usually provided via a roll forward mechanism, which involves applying the logged changes to a copy of the database.

The capabilities of such systems, provide us with a framework within which we can manage GIS data, both cartographic and geographic, allowing multiple users to access a single continuous model of the area of interest.

DATA ANALYSIS OF CARTOGRAPHIC FEATURES

How then to model the geometry of cartographic features? Fortunately, paralleling the development of database management systems, methodologies have been developed to analyse data, its structure and its integrity constraints. We have used one such methodology to analyse cartographic data, in much the same way we would analyse any other type of data. The result is a datamodel (Figure 1) that represents the data entities forming cartographic images, and the relationships between them.

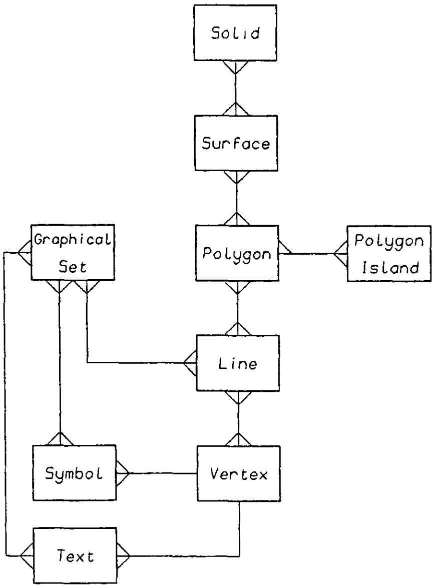


Figure 1.
Datamodel of
cartographic entities
(not normalised)

The following is a brief description of each entity and the relationships it has with other entities.

- The SOLID entity represents a solid region, formed by a closed set of surfaces. Any number of surfaces may be used to form a solid, and conversely a single surface may be common to more than one solid. hence there is a many-to-many relationship between solids and surfaces.

- The SURFACE entity represents a single continuous surface and is formed by a set of planar facets (planar polygons). Any number of facets may be used to form a single surface, and a single facet may be shared by multiple surfaces. Consequently there is a many-to-many relationship between surfaces and polygons.

- The POLYGON entity represents a closed area formed by a set of lines. Any number of lines may be used to form a closed polygon, and conversely, a single line may form part of the boundary of more than one polygon. Hence there is a many-to-many relationship between polygons and lines.

- The LINE entity represents any type of line segment, be it a straight line, an arc, a smooth curve or a line string with many vertices, that is unbroken by any other line possessing the same associated attributes. A line is formed by a set of vertices defining its position in space, and conversely a single vertex may be shared by more than one line.

- A VERTEX identifies a unique spatial location and possesses attributes that define that location, ie. its coordinates.

- The SYMBOL entity represents a point feature such as a trig station, a man-hole, a transformer etc. Each instance of a symbol is located at a single spatial location, but more than one symbol could exist at that location. Hence there is a many-to-one relationship between symbols and vertices.

- The TEXT entity represents free standing text. Normally in the GIS environment, text is stored as an attribute of a geographic entity. However, this feature has been included for users of data obtained from other systems (e.g. Ordnance Survey), that support free standing unassociated text.

- A polygon may contain other polygons (islands or holes), and they may be nested to any depth. This is a recursive relationship and can be represented in a number of ways. Here we define a one-to-many relationship between a polygon and the polygons it contains.

- The GRAPHICAL-SET entity is used to relate a set of cartographic entities that need to be referenced as a single entity. Examples include formed road boundaries, all the elements required to form a river or an airport etc. A graphical set may itself contain other graphical sets, lines and symbols.

It is important to note that the familiar "node-link-polygon" structure (by whatever name we choose to give it!) appears in a similar manner here. A significant difference to most other formulations however is that the vertex is the only entity to directly contain coordinate information.

Using this data structure, data is not duplicated, since each vertex, and line is only stored once. Consequently, moving a vertex moves the endpoints of all lines referencing the vertex. Similarly, when dealıng with polygons, no duplicate lines are required since each line is shared by all polygons referencing it.

Before implementing this model in a relational DBMS we must first apply the procedure of "normalization". The objective of normalization is primarily to remove redundancies and produce a "clean" structure. The result is a new datamodel where the many-to-many relationships between entities are resolved by the introduction of "junction entities". This model may then be implemented directly, as each entity maps to a database table (relation).

A geographic feature will have a set of geographic attributes and an associated cartographic representation. In fact sometimes a single feature may have more than one cartographic representation, for example a city may be represented as a polygon at one scale and as a point symbol at another. This association is achieved by the geographic feature table containing a reference to each type of cartographic entity used to depict it.

DBMS GRAPHICS PERFORMANCE

An integrated DBMS approach where both geographic and cartographic data are managed within the same database could create severe performance problems. Normal interactive database applications typically require the DBMS to return to the user a screenful of information (alphanumeric) per transaction. Such requests require the display of very few database records/fields. Transactions which scan large numbers of database records are not usually considered as interactive tasks.

When a GIS application requests the cartographic display of a geographical area, there may be several thousand cartographic entities to display, e.g. lines, symbols etc. Consequently many thousands of database accesses may be involved.

Database management systems tend to be very disk intensive, requiring on average between 1 and 4 disk I/Os per random retrieval of a specified record, depending on the indexing method used.* Disk is a relatively slow device, with normal access rates in the range 30 to 100 seeks per second. Once on cylinder, data is transferred to the CPU at a high rate, but rotational latency and head positioning affect overall performance considerably. Competition for the device from other users could further reduce the effective seek rate for each user. Consequently, the time required to retrieve all the cartographic data for an area containing thousands of cartographic entities is potentially unacceptable unless some method can be found to reduce the number of disk accesses.

ONE APPROACH TO SOLVING THE DBMS GRAPHICS PERFORMANCE PROBLEM

Until now we have discussed "logical" database design, and possible performance problems associated with transactions on that database. The following is a discussion about "physical" database design options, which aim to solve the performance bottleneck.

There are several methods commonly used with hierarchic and network DBMS's to improve performance that involve grouping together database records of a particular type. When a request is made to read a particular record from the database, the address of the record is determined, and a request is sent to the disk subsystem to retrieve the block or sector within which the required record resides. Many records may reside in the block of data returned, and the DBMS is responsible for extracting only the required piece. The entire block may then be held in an area of memory called the "cache". If at a later time the user requests the same record, or a record within the same block, it may be retrieved directly from cache, rather than from disk. The system simply checks whether the required block address matches any of those in cache and acts accordingly. Consequently, if records can be grouped in some way, then the number of I/Os could be reduced, improving response.

DBMS use a variety of techniques to provide fast access to a particular record within the database. Different techniques possess different attributes, with respect to how they distribute the data throughout the physical database. The hash technique is often used to randomise the data throughout the physical file, although it can also be used to group similar data together. In particular, a hybrid hashing technique has been suggested (Davis 1986) that can be applied

* For B-Tree indices, in the general case of retrieval of random records, from a table of R records, we would expect the number of disk I/Os to be $O[\log_n(R)]$, where n is the number of nodes held in a node block of the index.

to the organisation and indexing of spatial data. Alternatively, indexing techniques such as the B-Tree index, order the nodes of the index tree in ascending key order. If the data records are held in the leaf nodes of the tree, then they also will be ordered in ascending key value (certainly within each block and possibly partially within the file). Throughout the remainder of this paper I have assumed that B-Tree indexing is being used.

Normal GIS activities result in the user viewing a geographical area and performing transactions relevant to that area. If that area, represents a small fraction of the possible universe contained within the continuous model, then we can say that all the cartographic entities visible in that area must have "similar" coordinate values. Consider a function

$$s = f(R,R_u)$$

and $R = (r_x,r_y)$ is the range of the cartographic object

and $R_u = (r_{ux},r_{uy})$ is the domain of x and y

where $r_x = (x_{min},x_{max})$ is the range of x

and $r_y = (y_{min},y_{max})$ is the range of y

such that $f(R,R\ )$ returns a scalar, and that similar coordinate values yield similar function values. If this function is used to create values for the primary key of a cartographic entity in a B-Tree indexed database, then the probability of entities having similar coordinate ranges being in the same block on disk is increased.

Effectively, what we are attempting to do, is impose a one-dimensional ordering on a 2-dimensional space. Some of the techniques that may be used have been described (Mark et al 1986), and in particular the techniques of bit-interleaving of coordinates (Abel 1986), and quadtrees (Mark 1986) have been used to create primary keys for vector cartographic databases.

A modification of the quadtree technique has been used in a prototype GIS developed by Cambridge Interactive Systems Ltd. The quadtree technique can be thought of as a recursive subdivision of the universe until, in the vector case, further subdivision requires that the entity concerned be broken. In the raster case, subdivision continues until the cell is homogeneous. In each case the address of the cell, possesses both locational information and cell size information. The technique fails (for the vector case) when small entities cross large cell boundaries, preventing further subdivision of the cell. Overlapping cells may be used to overcome this problem.

In summary, the combined use of:

- cache memory

- an indexing technique that orders data, and

- a primary key value that is a function of geographical location,

can provide a mechanism whereby cartographic data may be retrieved from very large databases, with a substantial decrease in the number of physical disk I/Os required. Experiments (using these methods and a cache size of 32K words) have indicated that, typical GIS queries yield cache hit ratios of between 30:1 and 150:1. The ordering mechanism makes these results virtually independent of total data volume. Furthermore, the same approach can be extended and generalised for managing n-dimensional data, and non-cartesian coordinate systems, simply by selecting a suitable function to define the primary key.

OVERVIEW OF DATABASE DESIGN

A prototype GIS has been developed using the techniques described above. It utilises a relational DBMS to manage both geographic and cartographic data in a spatially continuous manner. The database is logically broken into four sections, cartographic, geographic, product dictionary and data dictionary.

The data structures required to model the geographic attributes of real world objects, will vary enormously from one organisation to another, and hence must be user defined. The relationship between a geographic entity and its cartographic representation must also be determined.

The Product dictionary manages data that describes the products that a GIS user may wish to create. It should be noted that, the same real world object, or more correctly its cartographic representation, may appear in a variety of GIS products, each with perhaps a different scale and symbology. For example, an airport may be represented by an aircraft symbol on a small scale map, with a generalised perimeter and different aircraft symbol on a medium scale map, and with detailed perimeter, runway and building information on a large scale map. The database should contain only a single representation, i.e. the most detailed level of data, and the scale and usage of the product should define the symbology. Hence in general, the appearance of the object must not be predetermined and stored with the cartographic entities defined previously. Instead, the symbology and generalisation requirements, should be associated with a definition of the product. In the prototype GIS, these requirements have been implemented as a set of symbolisation rules that may be associated with a set of cartographic products. The same technique has been used to define rules for the display of geographic attributes as cartographic text.

The data dictionary has been implemented as a set of tables within the database. It manages data describing the structure of the entire database, access rights, database users, and data to automate the creation of screen forms. A query language is being developed that makes extensive use of the data dictionary. The aim is to allow the GIS user to create, modify, and query the entities within the database in a manner that is natural to him/her, using the words that are normally used in that particular application area. In particular, we cannot expect a user to know the structure, and names of the cartographic entities in the database. Hence the query language via use of the data dictionary must support:

- aliases for both field and table names
- automatic recognition of the relationship between a geographic entity and its cartographic representation
- cartographic operators.

One further note concerning the query language is that it separates the retrieval of data from the display of data, contrary to the more normal SQL SELECT statement.

EXAMPLES OF APPLICABILITY

The prototype GIS has been developed using BaCIS II, a high level, polymorphic, procedural language developed by Cambridge Interactive Systems Ltd. The GIS environment integrates pure database access, interactive graphics, the query language, menu subsystems, and BaCIS II.

Tests have been performed in a number of GIS application areas, including:

- cadastral, electoral, administrative
- facilities management (water and electricity)
- small scale topographic
- mining and modelling of mine waste deposition
- local authority applications.

The topological structure of the database, has enabled users to perform complex network and polygon interrogation with ease, while the data dictionary provides the ability to interact with the model in the users own jargon. Graphic interaction is currently via 2D images, although 3D solids may be viewed and shaded images created.

CONCLUSION

Further development is continuing, but the feasibility of this approach has been proven. The ability to model the world in a continuous form, eliminates many of the problems associated with traditional mapping systems, and the integration of DBMS technology takes us beyond computer aided cartography into the realms of an Information System.

REFERENCES

Abel, D. J., (1986), Bit-interleaved keys as the basis for spatial access in a front-end spatial database management system: Proceedings, Volume 1, Auto Carto London, pp. 163-177

Date, C. J., (1986), Relational Database (Selected Writings), Addison-Wesley Publishing Company Inc.

Davis, W. A., (1986), Hybrid Use Of Hashing Techniques For Spatial Data: Proceedings, Volume 1, Auto Carto London, pp. 127-135

Mark, D. M., (1986), The Use of Quadtrees in Geographic Information Systems and Spatial Data Handling: Proceedings, Volume 1, Auto Carto London, pp. 517-526

Mark, D. M., and Goodchild, M. F. (1986), On the ordering of two-dimensional space: Spatial Data Processing using Tesseral Methods, (collected papers from Tesseral Workshops 1 and 2), Natural Environment Research Council, pp. 179-192

Developing a DBMS for Geographic Information: A Review

Gerald Charlwood
George Moon
John Tulip

WILD HEERBRUGG LTD
513 McNicoll Avenue, Willowdale
Ontario, Canada M2H 2C9

A summary of the development of a database for a geographic information system. The commonly described disadvantages of the relational model (fixed length fields and an excess of tables) were overcome in a variety of ways, allowing the retention of the advantages of the model. The Binary Data Model (BDM) was used to define the system specifications. A software tool was developed to convert the BDM specification into tables in a relational model and into an object oriented interface to the relational database. A small, dedicated development team followed a strict development cycle, resulting in all major milestones being met. One of the main themes in this paper is the handling of complex (spatial) data that does not obviously suit the relational model.

Introduction

Your mission, should you be so bold, is to construct a database to handle highly structured, multi-purpose geographic information and associated textual data, with display capability, and hooks to independent databases, for huge quantities of data to be randomly updated, with some real-time insertion.

This paper presents a review of the authors' experiences while developing a relational database with support tools to handle both spatial and non-spatial data. [3] describes the concept of the system we have developed. It is hoped that this paper provides a useful narration of events in a moderately large geographic information system undertaking. We review the database history from the definition of the original goals, through design, and development, to optimisation. At the time of writing, the database and tools are still being tuned, with initial feedback from first customers. One of the main themes in this paper is the handling of complex (spatial) data that does not obviously suit the relational model.

How did we decide to proceed? What worked? What didn't? This paper follows the actual development cycle.

A number of key decisions may be identified, which, with the chosen hardware and software, defined our working environment.

The product uses a commercial relational database management system (dbms) on a network of Sun-3 (tm) workstations running UNIX (tm) 4.2. Components of the system are linked with a proprietary inter-process communication protocol.

UNIX, while exceedingly popular, has disadvantages. The caveats on the choice of UNIX are that it is not a real-time operating system, and that it is not optimised for the needs of our application (e.g. job scheduling algorithm). In particular, there is a prejudice against processes requiring large amounts of memory and remaining active for long periods of time. UNIX does provide an available, portable, proven environment with excellent system development and support tools.

As mentioned above, we opted for a relational dbms to run under UNIX. In fact, we use the relational dbms not just to produce a particular database but to produce a custom database management system for our customers to develop their own databases. We provide definitions and support structures for the types of data anticipated in our target applications. It is then up to the end user to define the classes of things desired in their application (houses, roads, utility networks, waterways,...).

The relational dbms gives us the flexibility required to support a host of diverse applications. The relational algebra governing the query language is simple and powerful for end users. The reader is referred to [1] for more information on the relational dbms. As with any approach to any non-trivial task, our path was not without its difficulties. Two standard criticisms of the relational model for spatial data are that relations are implemented as tables with only fixed length fields, thereby wasting a lot of space, and that the large number of tables required in a normalised database (db) is hard on performance. These two problems come together in the fact that each 1:many or many:many relation must be implemented as a separate (binary) table. The relational dbms we use supports both variable length text fields and variable length fields of undifferentiated content ("bulk"). The former allow us and end users to store variable length text without wasting space. More structured information, such as lists of coordinate triples or pairs, can be put into bulk fields, with no wasted space. This addresses the first criticism, in that there is no restriction to fixed length fields. The second criticism is partially addressed also, since information that would otherwise require new tables can be put into bulk fields, so long as there is no need to use the relational algebra. Further handling of this performance question will be described below.

A number of alternatives to our approach exist in the market place. These include the use of a proprietary file structure with no dbms, some proprietary files with a dbms, and use of a dbms without variable length fields. We find that the costs of abandoning the dbms: losing the report writer, transaction logging, security, recovery, and rollback are too great. These same drawbacks arise, to a lesser extent, if a dbms is used with some proprietary files. The use of a dbms without variable length fields was felt to be too wasteful, as noted above.

We found two development paradigms to choose between: requirements driven, top-down, structured design and development or rapid prototyping with a quick turn-around time between prototypes. We opted for the former, although our requirements were incomplete, controversial, mutable, and inconsistent (i.e. normal). As we were not developing the db software in a vacuum - other members of our development team needed tools to work with - we made prototypes available for internal use as quickly as possible. One impact of this necessity was that developing the range of functions was more important than performance for our internal product. We evaluated performance along the way however, with an eye to future improvements. The contents of the early prototypes were the data components we believed to be necessary for our product. These components mainly involved the storing and retrieving of large amounts of topographic data. Graphics support data was added later.

We decided to have a small, tight group build the database and tools, as opposed to a large, shifting or distributed group. The rationale was to create a team atmosphere where intimate working relationships would foster a smooth flow of ideas and mutual assistance.

Contents and Queries

Each geographic database contains a mixture of spatial and non-spatial (mostly textual) data including definitions of the spatial and attribute data to be captured and manipulated, on which a wide variety of queries need to be supported.

The basis of the spatial data is spatial primitives of various topologic types: node, line, and surface. On these are built simple features and triangles. Complex features are built on simple ones. Interactive assistance is provided for defining the structures of simple and complex feature classes customers require.

That is, the spatial data are organised:

- primitives
  - [1] nodes
  - [2] simple lines
  - [3] arcs
  - [4] smooth curves
  - [5] circles
  - [6] surfaces
- features
  - [1] simples
  - [2] triangles
  - [3] complexes

Non-spatial data include:

[1] attributes - text, character, (long) integer, or floating point - of the various primitives and features,

[2] references tying the primitives and features together, sometimes taking the form of distinct tables, and sometimes variable length fields of either text or bulk,

[3] references to attribute data in other databases, which may be external to our system,

[4] definitions of feature classes,

[5] the apparatus to support graphic display, and

[6] support for database management.

Database management depends on the organisation of data into databases called projects, with working subsets, also databases, called partitions which must be checked in and out of projects. This provides a central repository of data (the project) with the capability of multi-user access and update via the various partitions.

In general, each captured piece of spatial data is stored once and may be displayed in a variety of ways, with user selection of which other data is to be displayed. Thus data content is distinct from data display. Selection of data to be displayed is done when the partition is defined. This selection is done by choosing a number of "themes". Each theme specifies classes of data to be displayed, a scale, and graphic attributes for each class. Thus each theme provides a way of displaying a subset (possibly all) of the spatial and attribute data in the partition. Distinct themes may display different data or the same data in different ways.

An issue arising from the complexity of the data structures involved is the management of shared primitives and features. Sharing of primitives and features arises when the flexibility of the data structure allows two or more spatial entities to build on the same primitive or feature (e.g. a road and cadastral parcel might share a boundary linear primitive). If a shared primitive or feature is moved or deleted, all the features referencing it must be identified and updated. Advantages to allowing sharing are that there is a saving of space, and that when a shared primitive is edited, all features referencing it are, in effect, edited. Thus, if a river is a national boundary and the river moves, it is not necessary to also update the national boundary. In cases where two features are desired to be contiguous, but only accidentally, it is easy for the user to create them using no shared primitives. The possibility of shared primitives showed up clearly in the data model and was approved by marketing and users.

The query language sql (tm) is supported by the relational dbms, taking advantage of explicit database structure. There is here a balance to be maintained between the pull of performance which tends to hide structure and the pull of the query language which uses it. For example, coordinate lists for lines may be stored in bulk fields, reducing the number of tables required. On the other hand, standard sql is only of limited use for these lists. We found it useful to extend sql by adding grammar and vocabulary to handle referencing between spatial entities, to handle queries based on the values in bulk fields, and to handle spatial relationships such as overlap, connectivity, and containment. For example suppose we want to select the names of all hospitals in Dover in the Kent county partition in project England database. Note the method of identifying the project and partition in the queries. Assume that "hospital" and "town" are (user-defined) feature classes. Classes town and hospital have defined attribute "name". That is, each town and hospital may have a name. (The user specified, during the definition of the project, whether the name is mandatory and its maximum length.) The first query assumes that each hospital is stored with an attribute "town_name".

```
Select hospital.name from England(Kent)
        where   hospital.town_name = "Dover"
```

If the town name is not available, we can retrieve the hospital names by looking for hospitals spatially contained within Dover. This uses the fact that, in the system, every spatial object has a stored minimum enclosing rectangle ("mer"). This uses an embedded select: first get the mer of Dover, and then compare it with hospital mer's.

```
'><'  signifies spatial containment

Select hospital.name from England(Kent)
        where   hospital[mer] ><
                [select town[mer] from England(Kent)
                        where   town.name = "Dover"]
```

The other direction of extension of sql is in the handling of data in bulk fields. Selection is supported on values of data elements within structures in a bulk field, and based on the ordinal position of the structure in the list of structures in the field.

For example, it is possible to select lines where the x coordinate of a structure in the coordinate list for the line is greater than (less than, etc.) a given value. That is:

```
select lines from England(Kent)
        where line[coord.x] > 100.0
```

It is also possible to select lines where the first (second, third, etc.) coordinate has a y value satisfying some condition.

Design

We will not deal in this paper with the general question of the architecture of the system, except to say that it is "modular": in working with the system a number of processes must cooperate, communicating with one another. The structure of the database part of the system is presented, along with a description of the methods and tools used in its development.

Five principles governed the design of the interface to the database.

[1] It must be object oriented: presenting objects intelligible to the end user, with components describing the object′s properties and relationships. Objects are described in detail below.

[2] It must shield the users, both programmers and users of the query language, from the underlying tables.

[3] It must use generic low-level update routines to minimise the effort and time involved in development.

[4] It must provide a consistent interface to the data. This interface should use a limited number of routines rather than one routine for each data element. In addition, application programmers should be able to go to a single source to discover the definitions of the objects. These definitions are contained in the TG input (see below). Along with a list of all the objects and all their components, is a description of the data format of each component, with the relevant constraints. These constraints include, but are not limited to, whether the component is mandatory and whether it is read only, write once, or repeatedly writable.

[5] Use of a memory cache of objects would minimise file I/O. This cache should contain the data being actively used by the application. The latter can access data in the db tables (relatively slow) or in the cache (fast).

The db and management tools may be viewed as a layered whole with the relational dbms at the heart. This is surrounded by a layer of utility functions to handle variable length lists, an object cache, data dictionary routines, and generic, db-internal read/write/delete functions. The use of these generic routines is crucial since they rely on generated code and handle almost all cases uniformly. Around this is a layer of application read/write/delete functions, functions to manipulate objects in the cache, and functions to create, delete, move, open, and close databases. This layer provides the consistent, concise application interface. Around this is the world of application programs and the extended query language.

Application layer:	applications, queries
Application interface:	read, write, update routines, cache object functions, db functions
db internal layer:	variable length list functions, object cache, data dictionary functions, generic read, write, update routines
kernel:	relational dbms

From the point of view of an application accessing a database or of an end user, the database contains spatial objects such as houses, roads, nodes, lines and non-spatial objects such as database definitions, graphic transform definitions (for defining display characteristics), and themes.

In general terms, an object is characterised by its properties and its relations with other objects. Its properties include things like its identification number, its class identifier, its name, its minimum enclosing rectangle, or its description. Possible relations include that fact that simple features reference primitives, surface primitives reference lines (and perhaps other surfaces), complex features reference simpler ones, partitions are owned by projects, themes are used by partitions, and graphic transforms are associated with simple features and primitives, given a theme. Note that the latter is a ternary relation (theme + class = graphic transform) which is easily handled in our data model, while causing difficulties for the entity-relationship model. These properties and relations are realised in an object's "components", which may be of fixed or variable length. Objects give the application programmer, and the query language user a view of the data which is independent of the particular tables involved, and therefore of changes to the underlying implementation. This becomes essential during performance tuning and when there are changes to the BDM specification.

Two tools are instrumental in the design, development, and evolution of this multi-level object: the Binary Data Model ("BDM") [2,4,5] and the table generator ("TG").

In brief, the BDM is a way of handling metadata: a method of analysing, organising, and presenting information handling requirements of a database. It enables system designers to work with end users to agree on a mutually comprehensible specification of the database contents. It is accepted by ANSI/SPARC as the standard for abstract data modelling. From this specification it is a simple algorithm to arrive at tables for a relational database in at least third normal form. The BDM rivals the entity-relationship model, but is more expressive and more readily yields a database implementation of the specified structures.

Results of analysis of the database requirements are expressed in a language which may then be used to produce graphical portrayal of the analysis and to produce input to TG.

Given this input, TG produces a specification of database tables, objects, and mappings between these two views. The generic read/write/delete functions rely on these mappings. Thus, we have an automated environment which goes from a "user friendly" specification of the database contents to database tables, object definitions, and functions mapping between tables and objects.

Advantages, to the end users and developers, of this approach include:

[1] The initial specifications are intelligible to end users and function as computer input.

[2] TG eliminates human error in generating tables, objects, and functions from the BDM specifications.

[3] It is easy to re-run TG whenever the initial BDM specifications change.

[4] TG guarantees that the same algorithm will be consistently applied to generate tables and objects. (People do move on.)

[5] Guaranteed consistency in data representations: if one element of the initial specification occurs as fields in several tables, or as multiple fields in one table,

we are guaranteed that each occurrence of it has the same data format.

[6] The generic read, write, and update routines greatly reduce the amount of code to be produced, thereby reducing costs and shortening the schedule.

[7] The insulation of the applications from the underlying tables makes possible various performance enhancements, without having to rewrite all the applications.

TG and BDM together are an invaluable time-saver, in addition to contributing to the internal consistency of the product and ensuring that what the user saw is what the user will get.

Performance Considerations

Having produced an initial version of the product, having shown the objects and functions described above to be feasible, we turned to performance issues.

There are four areas to look at: profiling of code execution to determine critical modules, attention to inter-process communication, minimising disk I/O, and minimising file I/O.

Rather than spending a lot of time during development trying to optimise all the code and algorithms, profiling of in-house test code and applications was used to determine the bottlenecks. Having found the slow points in execution, there are various remedies. Sometimes it is found that code is superfluous, perhaps because an integrity check is being done twice. Sometimes it is found that an algorithm can be improved upon: perhaps it was originally too general or simply not the best available for the task. The slow points discovered in code execution included:

- interrogation of internal data structures used to convert objects to db tables. The solution was to change TG to generate different mapping structures which support faster access to the database.

- the functions for handling variable length lists. Mechanisms were implemented to force more of the lists to remain in memory.

- updating indices when adding significant amounts of data. It is much faster to drop the indices during update and recreate them afterwards. This assumes that the data has enough integrity to guarantee that there will be no violations when re-creating unique indices. Facilities were provided to allow indices associated with objects to be dropped and recreated.

- queries on the object cache. Cache queries were accelerated by implementing an internal indexing scheme and by modifying the cache organisation.

- Spatial retrievals from the dbms. These are now performed by accessing an internally developed spatial indexing scheme. The indexing method is based on two dimensional extendible hashing. Initially, the indexing software made calls to the variable length list handling functions. This was found to be too slow and was replaced by a layer of software which manages the index directly. Pages from the extendible hash are now cached directly in a memory area of fixed size, and swapped on an lru (least recently used) basis.

Performance increases due to code optimisation ranged up to thousands of percent in some parts of the system. Overall performance has increased by a factor of ten as compared to the initial prototype.

Note that the extended query language is not affected by these changes since the query language software gets data from the db using the application interface layer of the db and is immune to changes to the underlying structures.

For the future, a number of possible paths exist. Two of these involve further reductions of file and disk I/O. The first of these is that cached objects may be stored in a new database, using the bulk fields, with many fewer files than the original. On this approach, we could reduce the number of tables to one, or to the number of object types supported. The basic table layout would consist of a primary key section followed by a data area: the objects would be linearised and stored in bulk fields. One issue here is handling of updates: objects store duplicates of information, unlike normalised tables. Another route would be to develop a table management and caching scheme to reside on top of the commercial dbms. In this scheme, we would map many records into a single relation managed by the dbms vendor. We would be responsible for getting the correct data out of the single relation. The mapping could be based on pages of records. This is not a trivial amount of work. In either case, the object cache manager would be changed to use a cache of fixed size, instead of the present, virtually infinite cache. The cache manager would be responsible for swapping objects or table pages in or out of the cache. A prediction algorithm could be used to ensure that desired pages are in memory as often as possible.

The db bottlenecks we found were, for the most part, very standard, arising from inefficient algorithms and data structures, I/O and the number of tables in the db.

The former problem was dealt with by modifying TG to produce more efficient structures, modifying internal routines to handle these new structures, and by redesigning the object cache to allow fast access to objects in core. It is noteworthy that only the internal routines had to be changed.

The latter problem was dealt with by reducing the number of reads/writes into the relational dbms through better utilisation of the object cache, and by replacing the calls to the variable length list functions with a layer of software to manage a cache of pages of extendible hash indices.

The overall modular architecture made it easy for us to juggle the number of processes and the grouping of functionality into various combinations of processes.

The UNIX 4.2 scheduling algorithm has a bias against large processes. On the other hand, inter-process communication can be a bottleneck, depending on the frequency and size of the information packets being transmitted. A balance must be struck among creating a large number of small processes, creating a smaller number of (large) processes and making efficient use of shared memory for inter-process communication. Initially, our design called for our database software and application software to run as separate processes, with our own inter-process communication software linking them. As there is a huge amount of traffic between such pairs of processes, it was found to be better to combine them.

Conclusions

The standard objections to use of a relational model for spatial data are the performance degradation due to the large number of tables involved and the need to use fixed length fields which waste space. These come together when handling line coordinate lists: either use a fixed length field, of virtually infinite size and waste a lot space, or save space, at the cost of another table and one table access for each coordinate in the list.

The latter objection is simply outdated. Relational db managers are being extended to support tables with variable length fields. Use of a fixed number of fields is not a commitment to fixed length fields. Variable length fields are useful for storing information about an object (e.g. the coordinate lists of lines), for information between objects (referencing information) and for storing whole pages of data.

The problem of the number of tables required is addressed by either linearising objects and placing them in bulk storage - so long as the problem of duplicated data is handled - or by implementing a proprietary table management scheme which would sit on top of the existing dbms.

While we obtain the advantages of a dbms, including transaction logging, security, and rollback, we can use variable length fields of text or bulk to avoid the problems inherent in a strict relational model without variable length fields.

The close-knit database development team met all its major milestones, and adapted well to shifts of direction and the changes required in tuning performance.

The use of the binary data model gives us a precise specification which users can evaluate, so there are no surprises when the system is delivered. Its use with the table generator gave us the ability to respond quickly and easily to changes in requirements: eliminating the need for repeated hand-crafting of huge amounts of crucial code. In addition, TG guarantees, within the limits of its algorithm, that what was specified in the BDM is what is built. The use of the binary data model and TG greatly enhanced the group's ability to supply the needed functions.

A modular, multi-process architecture allows us to optimise our use of the underlying UNIX environment by using a reasonable number of moderately large processes, with a balanced amount of inter-process communication.

Acknowledgements

We would like to acknowledge the support of Wild Heerbrugg, Heerbrugg, Switzerland during the preparation of this paper which is published with their permission. Details expressed herein are subject to change without notice.

Bibliography

[1] Date, C.J. 1983 - "An Introduction to Database Systems: Volume II", Addison-Wesley

[2] Mark, L. and Roussopoulos, N. 1986 - Metadata Management: Computer December 1986, volume 19, number 12, pp. 26-36

[3] McLaren, R. and Brunner, W. 1986 - The Next Generation of Manual Data Capture and Editing Techniques - The Wild System 9 Approach: Proceedings 1986 ACSM-ASPRS Annual Convention, volume IV, pp. 50-59

[4] Nijssen, G.M. and Carkeet, M.J. A comparison of Conceptualisation and Normalisation: University of Queensland

[5] van Griethuysen, J.J. (ed) 1982 - Concepts and Terminology for the Conceptual Schema and the Information Base: publication number ISO/TC97/SC5 - N 695.

GEOGRAPHIC INFORMATION PROCESSING IN THE PROBE DATABASE SYSTEM*

Frank Manola, Jack Orenstein, Umeshwar Dayal
Computer Corporation of America
Four Cambridge Center
Cambridge, Massachusetts 02142

ABSTRACT

This paper describes the facilities of PROBE, an object-oriented DBMS being developed at CCA (Dayal 1985, Dayal and Smith 1986). and how these facilities apply to the requirements of geographic information processing

INTRODUCTION

The application of database technology to new applications, such as geographic information systems, CAD/CAM, software engineering, and office automation, is an extremely active area of database research. The characteristics of these applications impose a number of requirements on supporting database systems that are "unconventional" when compared to requirements associated with conventional commercial database applications. For example, the requirements of conventional applications can be adequately modeled by relatively simple and regular .data structures, such as tables (relations), and a small, predefined set of operations on those structures On the other hand, GISs must deal with objects, such as features, that have complex and irregular structures Moreover, these objects will be created and operated on by a complex set of processes Such objects cannot be easily modeled by collections of simple attribute values, or by uniform database operations such as reads and writes Also, unlike commercially-oriented DBMSs which maintain only one up-to-date version of any record, a GIS may be required to maintain many different "versions" of information about the same object (e g , information at different scales, in both raster and vector forms, compiled at different times) Finally, commercially-oriented DBMSs are able to be fairly rigid in the class of data structures and processes on them that are supported, forcing users to adapt their requirements to these capabilities GISs have a greater requirement for adaptability, both to accomodate the diversity of specialized spatial data structures currently used in various applications, and to accommodate changes in technology and data requirements over the system life-cycle

In response to these requirements, a number of DBMS research organizations are pursuing development of "object-oriented" DBMSs (Dittrich and Dayal 1986, Lochovsky 1985, Manola and Orenstein 1986) These DBMSs allow meaningful application objects to be more-or-less directly modeled in the database Objects are accessed and manipulated only by invoking operations meaningful to the application, and specifically defined for the type of object involved Data structures and details of the implementation of the objects and operations can be hidden, or revealed only to specialized processes as needed Moreover, new object types, with their own specialized operations, may be freely defined by users for their own applications, rather than having to rely only on predefined, built-in data structures and operations PROBE is an object-oriented database system being developed by Computer Corporation of America This paper describes the basic features of the PROBE database system, and shows how they apply to GIS requirements The paper begins by describing the PROBE Data Model (PDM), and its spatial data capabilities This data

*This work was supported by the Defence Advanced Research Projects Agency and by the Space and Naval Warfare Systems Command under Contract No N00039-85-C-0263 The views and conclusions contained in this paper are those of the authors and do not necessarily represent the official policies of the Defense Advanced Research Projects Agency, the Space and Naval Warfare Systems Command, or the U S Government

model is an object-oriented extension to a data model called *Daplex* previously developed and implemented at CCA (Shipman 1981). The utility of Daplex in spatial data modeling has been described in (Norris-Sherborn and Milne 1986). The paper then describes our approach to incorporating spatial data processing into database operations, and other aspects of the PROBE system. The paper concludes by describing the current status of the system.

PDM DATA OBJECTS

There are two basic types of data objects in PDM, *entities* and *functions* An *entity* is a data object that denotes some individual thing The basic characteristic of an entity that must be preserved in the model is its distinct identity. Entities with similar characteristics are grouped into collections called *entity types* For example, a GIS might have an entity type **FEATURE**, representing geographic features

Properties of entities, relationships between entities, and operations on entities are all uniformly represented in PDM by *functions* Thus, in order to access properties of an entity or other entities related to an entity, or to perform operations on an entity one must evaluate a function having the entity as an argument For example

- the single-argument function **POPULATION(CITY)** → **integer** allows access to the value of the population attribute of a **CITY** entity
- function **LOCATION(PT_FEATURE)** → **(LATITUDE,LONGITUDE)** allows access to the value of the location attribute of a point feature (note that a function can return a complex result)
- the multi-argument function **ALTITUDE(LATITUDE,LONGITUDE,MODEL)** → **HEIGHT** allows access to the altitude values contained in a digital terrain model
- function **COMPONENTS(FEATURE)** → **set of FEATURE** allows access to the component features of a group feature (such as a city).
- function **OVERLAY(LAYER,LAYER)** → **LAYER** provides access to an overlay operation defined for sets of polygons separated into different coverage layers

Functions may also be defined that have no input arguments, or that have only boolean (truth valued) results. For example.

- the zero-argument function **FEATURE()** → **set of ENTITY** is implicitly defined for entity type **FEATURE**, and returns all entities of that type (such a function is implicitly defined for each entity type in the database).
- the function **OVERLAPS(POLYGON,POLYGON)** → **boolean** defines a predicate that is true if two polygons geometrically overlap All predicates within PDM are defined as boolean-valued functions

In PDM, a function is generically defined as a relationship between collections of entities and scalar values. The types of an entity serve to define what functions may be applied with the entity as a parameter value There are two general classes of functions *intensionally-defined (ID) functions*, with output values computed by procedures, and *extensionally-defined (ED) functions*, with output values determined by conventional database search of a stored function *extent* (ID-functions may also involve the use of stored extents, in addition to computation.) References to all functions are treated syntactically as if they were references to ID-functions, even when a stored extent exists, rather than treating the various classes of functions differently However, particularly in the case of ED-functions, functions can often be evaluated "in reverse" i e , with "output" variables bound, to return "input" values (since both are available in a stored extent)

Entity types may be divided into *subtypes*, forming what are known as *generalization hierarchies* For example, one might define **POINT_FEATURE** as a subtype of **FEATURE**, and **RADIO_ANTENNA** as a subtype of **POINT_FEATURE**. As another example, the declarations

```
entity LAND_DIVISION
DESCRIPTION(LAND_DIVISION) → character
AREA(LAND_DIVISION) → POLYGON

entity OWNED_PARCEL isa LAND_DIVISION
OWNERSHIP(OWNED_PARCEL) → OWNER
```

define a **LAND_DIVISION** entity type having two functions, and a subtype **OWNED_PARCEL** having an additional function. Because **OWNED_PARCEL** is a subtype of **LAND_DIVISION**, any entity of type **OWNED_PARCEL** is also an entity of the **LAND_DIVISION** supertype, and automatically "inherits" the **DESCRIPTION** and **AREA** functions On the other hand it sometimes desirable that specialized versions of what appears to be the "same function" be available for different subtypes For example, one might wish to provide a general **SQ_MILES** function to compute the number of square miles in any 2-dimensional shape, but have different specialized implementations for various representations of those shapes

At the top of the generalization hierarchy, both entities and functions are members of the generic type **OBJECT**. In addition, the entity and function type definitions themselves are modeled as a collection of entities and functions, so that information in database definitions can be queried in the same way as database data.

Generic operations on objects (entities and functions), such as selection, function application, set operations, and formation of new derived function extents, have been defined in the form of an algebra (Manola and Dayal 1986) similar in some respects to the algebra defined for the relational data model Like the relational algebra, our PDM algebra provides a formal basis for the definition of general database operations. In particular, the algebra serves to define the semantics of expressions in our query language, *PDM Daplex*, involving functions and entities, such as:

```
for C in CITY, for M in MAP
  print(NAME(C)) where
    POPULATION(C) > 50000
  and SQ_MILES(AREA(C,M)) < 10
```

This object-oriented approach has a number of advantages for a GIS. First, geographic data can be dealt with at a level of abstraction appropriate for the processing involved, as suggested in (Claire and Guptill 1982) For example, some users may wish to operate only at the "feature" level, ignoring how a feature may be represented in terms of geometric entities such as polygons, or how these polygons may be represented in terms of lower level constructs, such as vectors nodes or chains or their physical encodings This is true even when the users wish to use selection predicates or invoke operations involving geometric properties of features, since the functions can conceal their access to geometric or lower level objects (a terrain model may, for example, be in either regular grid or TIN form without affecting the user's view of the **ALTITUDE** function). Second, the use of the functional syntax provides a smooth interface between stored data and computations in the model (for example, interpolation in a digital terrain model). These computations may include complex cartographic processes and knowledge-based techniques (which may be implemented by specialized hardware or software), even when these are not part of the DBMS per se, since the processes can be represented by functions in the model, and referenced within "database" requests Again the functional syntax allows the exact nature of required processing to be hidden when this is appropriate, and provides a uniform syntactic approach throughout the model

SPECIALIZATIONS FOR SPATIAL DATA

The general objects and operations provided by PDM do not obviate the need to define specialized spatial object types for specific applications, or the need to define the details of the implementation of spatial objects in terms of discrete representations in computer storage, and operations on them. They do, however, provide a framework within which these can be smoothly incorporated in an overall system architecture As suggested by the examples above, we model spatial properties of entities by functions (such as **AREA**) that map from the entities (such as a **LAND_DIVISION**) to entities of special types (such as **POLYGON**) that denote sets of points in space (such as lines, areas, or volumes), and embody various specific representations of such point sets. These special types are defined as subtypes of the generic spatial type **PTSET**, which represents a general set of points.

Only the most general point set semantics are defined for the **PTSET** type The detailed behavior and characteristics of spatial entities required for particular applications are defined in subtypes of **PTSET** defined to represent specialized classes of point sets (e g 2D and 3D point sets), and specific types of objects within those general classes (e g , 2D lines and curves, 3D solids) (Points and intervals can also be defined in the same way to represent temporal objects, thus allowing many different concepts of both space and time to be modeled) For each subtype, additional specialized functions are defined to represent the user-visible spatial or nonspatial properties, predicates, and operations appropriate to the type of object being represented. Moreover, "internal" functions, hidden from outside the object, are defined to represent aspects of the *implementation* of the object. Attributes, such as **ALTITUDE**, that vary in (2D) space can be represented by multiargument functions

For example, we might define subtype **POLYGON** as:

entity POLYGON
internal:
EXACT_POLYGON(POLYGON) → set of EXACT_REP
GF_POLYGON(POLYGON) → set of ELEMENT

Entities of type **POLYGON** store the actual representations of polygons. Subtypes such as **POLYGON** required for geographic applications would incorporate the mathematical constraints required to correctly represent a closed, bounded 2-dimensional shape, together with its topological relationships to other spatial entities The details of these representations are hidden from the general-purpose database routines within PROBE, but are used by specialized functions added specifically to manipulate polygons, and requiring access to the details of internal representations In this case, function **EXACT_POLYGON**, given a polygon, returns a set of entities defining the exact representation of the polygon, while function **GF_POLYGON** returns the PROBE geometry filter's representation of the polygon: a collection of elements resulting from the decomposition of the polygon (described in the next section)

Facilities for adding such specialized entity subtypes and functions are provided within PROBE These subypes would either inherit the definitions of operations from the **PTSET** type, or would provide specialized versions of such operations For example, if the intersection of two entities of the type **POLYGON** were expected to produce only common subareas (i.e , a result of the same type), a specialized version of the intersection operation would be required, since an ordinary point set intersection might also return common boundary segments (or points), or even shared components of the underlying representations. (Such dimensionally-constrained set operations are known as "regularized set operations" in the 3D solid modeling literature (Requicha 1980)).

In addition to dealing with **PTSET** entities as individual objects, there are many situations in which it is necessary to deal with PTSETs contained within other PTSETs For example, a map would be defined as containing a collection of component objects (roads land units, etc) The **POLYGON** defining the spatial representation of the map would have to contain all the **PTSET** entities (polygons and other representations) of those components (together with the complex topological relationships among them) Moreover, a component representing a complex feature may itself contain further subcomponents, and so on Thus, both the map and its component objects, and their associated spatial representations, naturally exhibit a hierarchical structure. While it is not possible to deal with this subject further in this paper, PROBE includes facilities both for modeling these containment relationships, and for efficiently processing queries involving them (Orenstein and Manola 1986, Rosenthal 1986).

By using different functions (or a single set-valued function), a database entity can be related to any number of different spatial representations For example, a bridge might be represented as a point in one map, as a line in a map showing greater detail, as a space frame in its design data, etc Also, as suggested above, the 2D point set representing the bridge in a particular map can be associated with the point set representing the area covered by the entire map, enabling the bridge to be associated (and located) with respect to the other features in the same map

QUERY PROCESSING

For a DBMS to be practically usable in a given application, it must provide adequate performance A key component of the DBMS in determining its performance is its query processor, since this is where efficient strategies for performing database operations are determined. A considerable literature exists on query optimization strategies for conventional database systems. However, because a conventional database system provides only a fixed set of data types and operations, support for those types and operations can be coded directly into the physical data structures and algorithms of the DBMS query processor This simplies the optimization problem considerably.

An extensible database system, such as PROBE, is more difficult to implement because it is no longer possible to build into the query processor's implementation the definition of each data type and operation that may be encountered. Instead, the extensibility of the system implies that some important details of the system will be provided in user-defined object types For example, for the query.

```
for S in STATE where NAME(S) = "Florida"
  for C in CITY where AREA(C) is in AREA(S)
    print(NAME(C));
```

the **AREA** functions might return values of a user-defined specialized data type, and the processing of the **is in** predicate might be implemented by a user-supplied procedure whose implementation details are hidden from the query processor

A reasonable division of labor would be for the query processor to handle sets of objects of the most generic types (e g , **ENTITY** and **PTSET**) and for the more specialized object types to provide for detailed manipulation of individual objects This allows DBMS implementors to be concerned primarily with generic database issues, while application specialists can concern themselves primarily with application-specific issues. For example, an application specialist could define a **POLYGON** object class whose operations work on individual polygons (e g. do two polygons overlap? does the polygon contain a point? what is the area of a polygon?) The DBMS can then implement operations that handle arbitrary numbers of polygons (e g find all the polygons overlapping a given polygon) in terms of these operations

Given this division of labor, how can the data model, especially its spatial components, be supported efficiently? Our approach to the problem has two components. First, we present an architecture compatible with the division of labor discussed above. Second, we describe how the efficient processing of spatial data can be accomplished under this architecture. This step is necessary to show that PROBE's approach to spatial data is feasible as well as general

Architecture for the Query Processor

Many queries (including spatial queries) can be expressed in terms of iteration over one or more collections of objects, and the application of one or more functions to each object or to a group of objects within the iteration For example, in order to find all pairs of objects in a set, S, of spatial objects, within a given distance, d, of one another, the following algorithm can be used:

```
for each x in S, for each y in S
 if distance(x,y) < d then output (x, y)
```

(It is a simple matter to eliminate symmetric results (x overlaps y iff y overlaps x) and reflexive results (x always overlaps x).) This kind of algorithm is very compatible with the division of labor discussed above The database system can handle the iterations, passing pairs of objects to a distance function (part of a spatial object class) whose boolean result indicates whether the pair should be part of the output This architecture is shown in figure 1

The problem with this approach is one of performance. Each loop over a set of spatial objects corresponds to an actual scan of the set Nesting these loops leads to polynomial time algorithms (whose degree is equal to the level of nesting). This will not be acceptable in practice since much more efficient, special-purpose algorithms often exist However, it is not possible to build in a collection of special-purpose algorithms and retain generality It is therefore necessary to consider another architecture

The PROBE approach is to provide a generally useful *geometry filter* that helps optimize such nested loops The output from the filter will be a set of *candidate* objects (or a set of groups of objects) that satisfy the query Any object or group that is not included in the candidate set is certain not to satisfy the query. An object or group in the candidate set *is likely to* satisfy the query. The set of candidates will then be refined to yield the precise answer by applying user-supplied predicates (such as distance). This architecture is shown in figure 2.

Note that this architecture is also compatible with the division of labor described above. The user only has to supply, as part of a spatial object class, a predicate that tests a group of objects. The geometry filter is part of the database system and relies on another object class (**ELEMENT** in figure 2) that is provided as part of PROBE The ideas behind the geometry filter and the **ELEMENT** object class are described below.

How the Geometry Filter Works

The geometry filter is based on a grid representation of spatial data. For a spatial object, s, a grid cell that contains any part of s is "black" while a grid cell that is completely outside s is "white" The collection of black cells forms a ***conservative approximation*** of s In order for the geometry filter to retain its filtering property (i e. not discard positive results), it is important for the approximation to be conservative - i e contain the exact representation This is because positive results are indicated by the overlap of objects A non-conservative approximation does not necessarily contain the exact representation and some overlap relationships involving the exact representation would not be detected by the approximation (Unless stated otherwise, all results in this section are from (Orenstein 1986, Orenstein and Manola 1986).)

Many spatial operations can be carried out in a single scan of a grid, replacing the nested loops algorithms described above The problem with this approach is that grids can be very large (at high resolution), and that, as a result, a scan of the grid will be very slow However, it is usually the case that spatial data contains much regularity There will be large black regions and large white regions The geometry filter exploits this regularity by using a compact encoding of these regions of uniformity. Each spatial object is represented by a collection of rectangular regions called "elements" Each element can, in turn, be represented by a range of integers. Typically, a 32-bit word is sufficient to represent an element

Elements are obtained by partitioning the grid in a highly constrained way As a result, elements have some simple and useful mathematical properties

- The size, shape and position of an element can be described very concisely by a "z value" - a short string of bits (Orenstein 1984) The same encoding has been discovered independently by several other researchers (Abel and Smith 1983, Gargantini 1982, Mark and Lauzon 1985)

- Any two elements either contain one another or precede one another (when ordered by z value). Overlap (except for containment) cannot occur.
- In a sequence of elements sorted by z value, proximity in the sequence is highly correlated with proximity in space.

These properties lead to simple and efficient algorithms for a wide variety of spatial problems. The scan of the grid cells is replaced by a scan or merge of z-ordered sequences of elements. Algorithms of this kind are possible because of the absence of overlap relationships.

The performance of these algorithms looks promising for two reasons. first, the property that proximity in z-value corresponds to proximity in space leads to good clustering. Database implementers go to great lengths to ensure that records to be retrieved together are stored near each other, ideally on the same page or cylinder (e.g., see (Lorie 1977)). In spatial applications, objects that are near each other are often retrieved together, and z order maps this proximity to proximity in real storage devices. Second, with some simple reasoning about z values, it is possible to "skip" parts of a space that could not contribute to the result. This reasoning has been incorporated into the geometry filter algorithms and there is analytical and experimental evidence that the savings are substantial. The performance for range queries matches that of the best practical data structures (e.g. the kd tree).

The representation used by the geometry filter, the collection of elements, can be seen as an abstraction of the quadtree and all its variants. There is an exact correspondence between a leaf of a quadtree and an element of the geometry filter. (The quadtree can be seen as a trie of order 4, keyed by elements. Similarly, the octtree can be seen as a trie of order 8, keyed by elements.) However, instead of requiring the use of a particular data structure, geometry filter algorithms can use *any* data structure or file organization that supports random and sequential accessing. This is a very important consideration since it permits the use of efficient and widely available structures such as sorted arrays, binary trees (and variations), B-trees (and variations), ISAM, etc.

The geometry filter supports a wide variety of spatial operations, including many operations for which the quadtree and related structures have been proposed. Given two sets of spatial objects, R and S, **spatial-join**(R, S) locates pairs of objects (r, s) such that r belongs to R, s belongs to S, and r and s overlap spatially. This operation can be used to evaluate range queries, partial match queries (important in database systems), containment queries and proximity queries. It has also been applied to interference detection and to polygon overlay.

EXAMPLE

(Zobrist and Nagy 1981) give several examples of multistep, geographic information processing tasks that demonstrate the need for manipulation, integration, and conversion of geographic data stored in different representations (without going into how the various steps might be implemented in a database system). To demonstrate the use of PROBE for geographic information processing, we now "translate" some of the processing from one of the examples into steps that could be carried out in PROBE.

The example selected is a study of the California Desert Conservation Area. We will concentrate on the latter part of the example, after multiple sources of data (such as LANDSAT frames and digital terrain data) have been integrated to yield, for each point of the study area, a classification. (Other steps in the example could be modeled in PROBE as well: for example, geometric transformations stored as PROBE ID-functions can be applied to LANDSAT images stored as PROBE point set entities; similarly vector/raster conversion routines stored as PROBE functions can be applied to boundary files stored as PROBE objects.)

To model the data and the required computations, we use the entity types **LAND_DIVISION**, **OWNED_PARCEL**, and **POLYGON** defined earlier, and additional types and functions such as

entity IMAGE
PIXELS(IMAGE) → set of PIXEL
COVERAGE(IMAGE) → POLYGON
CLASS(IMAGE,PIXEL) → CLASS_VALUE
QUALITY(IMAGE) → RATING

RASTERIZE(POLYGON) → set of PIXEL
PIXEL_TO_ACRE(integer) → ACRE

An entity of type **LAND_DIVISION** represents the area of study In this case, its **NAME** function would return "California Desert Conservation Area" An entity of type **OWNED_PARCEL** identifies a parcel of land whose owner can be identified. Since both are entities of supertype **LAND_DIVISION**, they have an **AREA** function that returns a polygon (assumed to have absolute coordinates) describing the land area **IMAGE** defines an image object, with a **CLASS** function that gives a classification value (vegetation, urban, etc) for each pixel for each image and a **COVERAGE** function defining its area of coverage. By specifying constraints on the values of image attributes such as **COVERAGE** and **QUALITY** a set of images can be identified **RASTERIZE** is an ID-function that, given a polygon, returns the set of pixels that are completely or mostly within the polygon (Pixels on the boundary are included in the result only if they are mostly within the polygon) Function **PIXELS_TO_ACRES** uses the unit of area covered by a pixel to convert from a pixel count to acres

The goal of the last part of the example is to overlay the land classification information with "boundary files" (giving ownership information in this case) to obtain acreages of land classes per region. This processing can be done in PROBE in five steps.

Select an image: There are several criteria that might influence the selection of an image The most important is the area covered by the image After selecting images that cover the area of the study, further selection can be based on other attributes To locate images in the right area, the following steps can be used

1. Do a spatial join between the **COVERAGE** function of the set of images, and the **AREA** function of the **LAND-DIVISION** entity representing the study area, to locate the images overlapping the study area
2. Select one image based on other attributes and create a new ED-function storing the needed information, **STUDY_IMAGE(PIXEL) → CLASS_VALUE**.

Get the relevant ownership information: Obtain the owned-parcel polygons, and do a spatial join with the study area's polygon to find the regions covering the study area.

Convert to common representation: Convert the owned-parcel polygons to raster format using the **RASTERIZE** function and produce a new ED-function
PIXEL_OWNER(OWNER) → set of **PIXEL**

Do the overlay: Each pixel is related to a region through the **PIXEL_OWNER** function, and has a class, as indicated by the **STUDY_IMAGE** function. These functions can be composed, yielding **OVERLAY(PIXEL) → (OWNER,CLASS_VALUE)**

Compute acreage of class per district: PROBE provides general-purpose aggregation functions, such as sums, maxima, minima, and counts In this example, aggregation can be used to count the number of pixels of each class in each region **USAGE(OWNER.CLASS) → count** Finally, the **PIXEL_TO_ACRES** function can be applied to **USAGE** to convert the count of pixels to area, measured in acres

CURRENT WORK

Work related to PROBE is ongoing in a number of areas, some of them mentioned in previous sections A breadboard implementation of PDM and its algebra, and of some query processing algorithms, is under way. The breadboard will be tested against a number of example applications (one of them a geographic application). This involves the definition and implementation of a number of specific entity types incorporating spatial semantics (Manola and Orenstein 1986).

ACKNOWLEDGEMENTS

We express our thanks to Alex Buchmann, Upen Chakravarthy, Mark Dewitt, David Goldhirsch, Sandra Heiler, and Arnie Rosenthal for their contributions to the development of PROBE.

REFERENCES

Abel, D J and Smith, J L 1983, "A data structure and algorithm based on a linear key for a rectangle retrieval problem", *Computer Vision, Graphics and Image Processing* 27(1)

Claire, R W and Guptill, S.C 1982, "Spatial Operators for Selected Data Structures", *Proc. Fifth Intl Symp. on Computer-Assisted Cartography*, ACSM

Dayal, U. et al. 1985, "PROBE - A Research Project in Knowledge-Oriented Database Systems: Preliminary Analysis", Technical Report CCA-85-03, Computer Corporation of America.

Dayal, U. and Smith, J.M 1986, "PROBE A Knowledge-Oriented Database Management System", in M L. Brodie and J. Mylopoulos (eds), *On Knowledge Base Management Systems Integrating Artificial Intelligence and Database Technologies*, New York, Springer-Verlag.

Dittrich, K and Dayal, U 1986 (eds), *Proc Intl Workshop on Object-Oriented Database Systems*, Washington, IEEE Computer Society Press

Gargantini, I 1982, "An effective way to represent quadtrees", *Comm ACM*, 25(12)

Lochovsky, F. 1985 (ed), *Database Engineering*, Vol. 8, No 4, Special Issue on Object-Oriented Systems, IEEE.

Lorie, R A. 1977, "Physical Integrity in a large segmented database", *ACM Trans on Database Systems*, 2(1), 91-104

Manola, F A and Orenstein, J 1986, "Toward a General Spatial Data Model for an Object-Oriented DBMS", *Proc 12th Intl Conf Very Large Data Bases*, IEEE

Manola, F.A. and Dayal, U 1986, "PDM An Object-Oriented Data Model", in (Dittrich and Dayal 1986)

Mark, D M and Lauzon, J P. 1985, "Approaches for quadtree-based geographic information systems at continental or global scales", *Proc Seventh Intl Symp. on Computer-Assisted Cartography*, ACSM.

Morehouse, S 1985, "ARC/INFO, A Geo-Relational Model for Spatial Information", *Proc Seventh Intl Symp on Computer-Assisted Cartography*, ACSM

Norris-Sherborn, A and Milne W J 1986. "A Practical Approach to Data Modelling in Spatial Applications", *Software—Practice and Experience*, Vol 16(10), 893-913, (October 1986)

Orenstein, J.A. 1984, "A Class of Data Structures for Associative Searching", *Proc. ACM SIGACT/SIGMOD Symp. on Principles of Database Systems*, New York, ACM.

Orenstein, Jack 1986, "Spatial Query Processing in an Object-Oriented Database System", *Proc 1986 ACM-SIGMOD Intl Conf on Management of Data*, New York, ACM

Orenstein, J and Manola, F. 1986, "Spatial Data Modeling and Query Processing in PROBE", Technical Report CCA-86-05, Computer Corporation of America

Requicha, A. 1980, "Representations for Rigid Solids Theory, Methods, and Systems", *Computing Surveys*, 12(2), 437-464 (December 1980)

Rosenthal, A. et al 1986, "A DBMS Approach to Recursion", *Proc 1986 ACM-SIGMOD Intl Conf on Management of Data* New York, ACM

Shipman, David 1981, "The Functional Data Model and the Data Language DAPLEX" *ACM Trans Database Systems* 6(1), 140-173

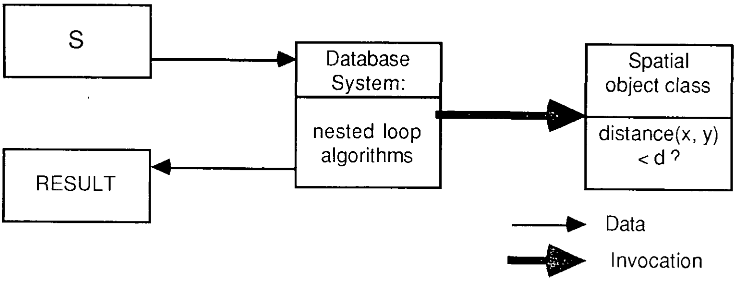


Figure 1.

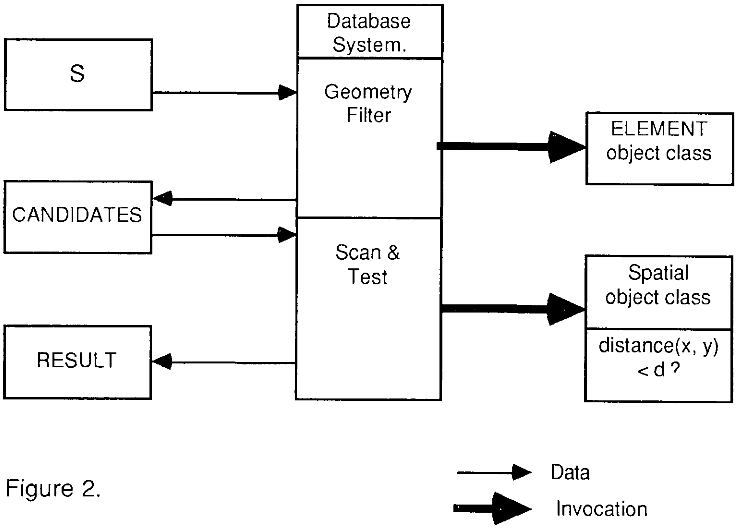


Figure 2.

GEOGRAPHIC INFORMATION PROCESSING USING A SQL-BASED QUERY LANGUAGE

Kevin J. Ingram
William W. Phillips
Kork Systems, Inc.
6 State Street
Bangor, Maine 04401

ABSTRACT

The utility of a Land Records or Natural Resource information system is greatly enriched if its geographic (or map-based component) and associated attribute data components can be integrated. Differences in how these two classes of data are now processed, in particular, the lack of a common query language, have impeded this integration. Kork Systems' new Geographic Information System (KGIS) combines separate geographic and attribute data bases into a single, integrated system. Geographic data (polygons, lines and points) are maintained in a fully-intersected topologic data structure with direct links to their associated attributes. The query language used in KGIS is based on the Structured Query Language (SQL). Several important additions were made to SQL to incorporate spatial concepts into the query language, including location, area, length, proximity and geographic context. These additions enable the user to form rapid, on-line queries about complex spatial relationships among the data.

INTRODUCTION

The utility of a Land Records or Natural Resource information system is greatly enriched if the spatial and non-spatial attribute components associated with geographic features can be integrated. Differences in how these two classes of data are now processed, in particular, the lack of a common query language, have impeded this integration. A query language is grounded in the data model on which it is based. In discussing the query language developed for Kork Systems' new Geographic Information System (KGIS), it is necessary to understand the data model on which KGIS is built. The remainder of this paper comprises a review of the data models available to a Geographic Information System (GIS) developer, a short overview of the data model used in KGIS and finally, a description of the query language used in KGIS.

DATA MODELS FOR GEOGRAPHIC FEATURES

The continuing development of GIS technology has involved two concurrent trends: 1) increasing sophistication of data models for both the spatial and non-spatial (attribute) information about geographic features, and 2) strengthening links between the spatial and non-spatial attribute portions of the information.

Data models for non-spatial attribute information have progressed from special purpose files designed to be accessed by specific application programs, through the hierarchical and network data models used in the first generalized data base management systems (DBMS), to the relational model which has emerged as a powerful and flexible structure for representing attribute information in tabular form.

The special characteristics of spatial data (Peuquet, 1984) have presented a challenge to GIS developers in their search for suitable data models. Early geographic data bases consisted of a catalog of files, segregated by map sheet and data layer, containing either vector or tessellated data.

Vector data usually consisted of polygon-digitized ("double-digitized") or spaghetti-digitized segments. Dueker (1985) has described the shortcomings of this data model. The topologic model, an improvement on earlier vector structures, made it possible to relate one feature to other features within a data layer. Relating data between layers, however, required an expensive polygon overlay operation.

Tessellated data, either grid-cell or raster encoded, provide for simpler computations at the cost of increased data volume and lower positional precision. The application of hierarchical tessellated data structures (quadtrees and other variants) to geographic data bases reduced the data volume burden and made spatial searches efficient.

The success of the relational model for managing non-geographic data has led to attempts (Waugh and Healey, 1985) to apply it to geographic data as well. This approach has the strength of storing the data in a sophisticated DBMS and the flexibility of the relational model. Since all information is represented by a single data model, this approach should provide a strong link between the spatial and non-spatial attribute components of the geographic information. It generally suffers by forcing the user to manipulate the information about geographic features at a very low-level (Abel and Smith, 1985).

The more recent development of the object-oriented data model has provided a powerful tool for managing geographic data, including model constructs such as classification, generalization and aggregation. One system (Frank, 1986) based on an object-oriented data model, PANDA, is implemented on a network DBMS and provides spatial access to geographic features through a modified quadtree structure for data storage. The object-oriented approach allows considerable knowledge about the behaviour of objects to be embedded in the system. However, this inherently limits the system's flexibility with regard to changes in the schema since any modification necessitates re-programming.

In summary, there does not yet appear to be a single data model for geographic features which is superior in every aspect to all other data models. As a result, the link between the spatial and non-spatial attribute components of geographic features has usually been either weak or non-existent. This shortcoming can be reduced by using a hybrid data model (Morehouse, 1985). However, if the hybrid nature of the data model is visible externally, then the user will be forced to work in one mode or the other. This tends to segregate specific capabilities by mode which reduces the systems overall flexibility.

KGIS OVERVIEW

In designing KGIS, we defined three functional requirements. First, the system should handle spatial and non-spatial attributes in a single context to give the user the maximum amount of flexibility in formulating queries. Second, it should have the ability to relate geographic features in one layer, e.g. soils, with those in other layers, e.g. bedrock geology or roads. Third, the system should not impose a fragmentation of the spatial data upon the user, but should maintain them as a seamless whole while permitting the user to define an arbitrary subset. To accomplish these requirements, we decided to distinguish between the external and internal views of the data, designing each view to support a different data model.

The external view supports the relational model, giving the user access to all information in the data base including both spatial and non-spatial attributes of geographic features. The user views geographic features at a high level, manipulating them in the same context as other information. Each thematic layer of geographic features is represented in a separate relational table.

Internally, KGIS is implemented on a hybrid data model (Keating et al, forthcoming). Non-spatial attribute data are maintained in a relational DBMS. The relational model provides the necessary flexibility in the schema. Spatial data are maintained in an object-oriented DBMS, PANDA. The object-oriented approach provides the high-level view of geographic features by hiding inner complexity of the data structure in the lower levels of the system. The data model for spatial attributes contains elements of several other data models as well. The schema defined for the spatial data stores them in a fully intersected topologic data structure, an improvement on the layer-by-layer topologic model. This schema permits a geographic feature in one layer to be related to features in other layers as easily as in the same layer. Finally, the PANDA DBMS provides a hierarchical tessellated data structure for storing and accessing spatial data. Therefore while no artificial fragmentation of the spatial data base is necessary, this capability enables rapid spatial searches of any arbitrary user-defined subset of the data base.

KGIS QUERY LANGUAGE FEATURES

SQL (Chamberlin et al., 1976) will soon be an official standard, query language for relational DBMSs. For this reason, it was chosen as the basis of the KGIS query language. However, SQL suffers from the same shortcomings as other commercial DBMS query languages when applied to geographic data management (Abel and Smith, 1985). Missing but necessary facilities include 1) a high level view of geographic features, 2) support for the graphic display of geographic features and 3) an ability to express spatial relationships as selection criteria. In addition, an adequate geographic query language (Frank, 1982) must provide support for 4) geographic context specification, 5) graphical context specification and 6) graphical input.

To provide these facilities, we have made several additions to SQL. Since maintaining compatability with the SQL standard is a priority, we have retained the overall syntax whenever possible, adding extensions as new commands or as functions for use with existing commands. The resulting language provides facilities in all of the areas described above.

High Level View Of Geographic Features

Geographic features are distinguished from other data base entities, which have only non-spatial attributes, by having spatial attributes and spatial relationships to other geographic features as well. By treating them as objects, the KGIS query language provides a high level view of geographic features, relieving the user from manipulating the complex internal representation of geographic information directly. Geographic features appear as entire entities, such as 'Parcel 123' or 'Route 1', which have both spatial and non-spatial attributes.

Graphic Display Of Geographic Features

The graphic display of geographic features involves two issues: 1) specifying the information to be displayed and 2) specifying the format in which to display it.

All geographic features have associated positional information which, when taken in its entirety, can be expressed graphically as a map. The map for a geographic feature, therefore, may be treated as one of its spatial attributes. To view the map of a feature, the user simply retrieves it like any other attribute. The following query

```
SELECT MAP FROM PARCELS WHERE VALUATION > 100000;
```

displays, on the graphics screen, a map of each parcel whose value is greater than $100000.

On an ordinary paper map, the legend describes graphic symbology used to represent the mapped information. Analogously in KGIS, a dynamic legend describes the symbology used to render the information displayed in map form on the graphics screen. This legend provides a mechanism for specifying how query results are graphically displayed. A default display format is maintained for each layer, e.g. parcels or soils, in the data base. This format provides a complete description of how a feature is to be displayed and it is referred to whenever a query involving graphics is executed. The legend is dynamic because, unlike a paper map, information can be added to or removed from the display and as this happens, the descriptions in the legend change correspondingly.

Spatial Criteria For Data Retrieval

A geographic feature differs from other data base entities in having spatial attributes such as size, shape and location, which depend only on the individual feature, and spatial relationships to other geographic features, such as proximity, adjacency and direction.

In addition to MAP, spatial attributes that are currently supported include AREA, PERIMETER and LENGTH. They appear to the user to be stored explicitly in the data base. In actuality, because of their dependency on a feature's positional information, these attributes are computed when they are requested. They are treated in the same syntactic context as other attributes. They may be retrieved along with a list of other attributes or used in selection constraints as the following example shows:

```
SELECT ID,MAP,PERIMETER,OWNERNAME FROM PARCELS WHERE
```

```
AREA > 10;
```

Spatial relationships between geographic features are much less tractable, often involving fuzzy or application-dependent definitions. Peuquet (1985) has stated that all spatial relationships appear to be derivable from three primitives: boolean set operations, distance and direction. Of these, direction is the least useful because a model for direction, free from dependency on human interpretation, has not been developed. As a result, we have focused our efforts to date on spatial relationships that can be derived from boolean operations and distance. Each of these imply, in a sense, a relational join operation, i.e. a spatial join. These spatial relationships relate two separate groups of features, or themes, over a shared domain, namely location is space. In certain cases the join criteria could be made explicit in terms of shared topology. We decided that this would overburden the user and opted instead to implement spatial relationships as high-level functions to better express a users intuitive understanding. All the spatial relationships we currently support fall into two classes: those that act like attributes and those that act like predicates. Attribute-like relationships include DISTANCE and OVERLAP. Predicate-like relationships include OVERLAY and ADJACENT.

DISTANCE is implemented as a scalar function of two themes. It expresses the minimum distance from a feature in the first theme to one in the second. It can be used either as an attribute or a selection criteria as follows:

```
    SELECT class, depth, map, distance ( soils, roads )
    FROM soils,roads
    WHERE distance ( soils, roads ) < 500:meters and
          roads.surface = 'Paved';
```

This query returns several soil attributes, including the map and distance to the nearest road, for soils which occur within 500 meters of a paved road.

OVERLAY is a boolean function of two themes. Stated as a predicate, it expresses the spatial intersection of a feature in the first theme with one in the second, e.g. polygon-polygon, line-line, point-in-polygon, line-in-polygon, etc. It is used as a selection criteria as follows:

```
    SELECT class, cec, permeability, soils.map
    FROM soils, parcels
    WHERE valuation > 60000 and overlay(soils, parcels);
```

This query returns information, including graphics, about soils which occur on parcels valued above $60,000 dollars. The topologic data structures, used in KGIS to represent geographic features, are built at the time the data are added to the data base and alleviate the need to perform polygon intersection computations at query time. Instead, the OVERLAY operation is reduced to identifying shared topology.

Often, with queries involving the OVERLAY relationship, information relative to the overlapping portion is required. Queries of this kind can be expressed using the OVERLAP modifier. OVERLAP is a scalar function of spatial attributes, used in a query involving the OVERLAY relationship to express attributes of the overlap. For example,

```
    SELECT r.id, overlap(r.length), overlap(r.map)
    FROM townships t, roads r
    WHERE t.name = 'Hampden' and maintenance = 'State'
          and overlay (roads,townships);
```

returns the id, surfacing material, length and map of State-maintained roads that pass through the township of Hampden. The OVERLAP function returns only that portion of the length and map which falls within Hampden.

ADJACENT is a boolean function of two themes. Stated as a predicate, it expresses whether the boundaries of two geographic features share a topologic 1-cell, referred to in KGIS as an edge. This relationship can exist between two polygons or between a line and a polygon. It is used as a selection criteria as follows:

```
  SELECT id, address, valuation
  FROM parcels,roads
  WHERE adjacent( roads, parcels) and
        roads.id = 'Elm Street';
```

This query returns information about parcels which are adjacent to Elm Street. As with the OVERLAY function, execution of the ADJACENT function consists of identifying shared topology between pairs of features.

Geographic Context

The locational data for geographic features is maintained in a single, seamless data base, not partitioned into pre-defined map sheets or their equivalent. If a user does not wish to query against the full geographic extent of the data base, a geographic context may be established. In keeping with the relational model, we refer to this geographic context as a GEOVIEW. A GEOVIEW effectively partitions the data base spatially so that only those geographic features that fall within the specified area are considered for retrieval. A GEOVIEW may use a geographic feature or an arbitrary ground window. A GEOVIEW is specified using a SET command as follows:

```
  SET geoview WHERE counties.id = 'PENOBSCOT'
                 or
  SET geoview WHERE lowerleft = utm(500000,3894000) and
                    upperright = utm(520000,3900000)
```

Once established, a GEOVIEW remains in effect for subsequent queries until changed or reset.

Graphical Context

To visually interpret query results displayed on a graphics screen, it is often necessary to supplement those results with background information. A base map provides a graphical context from which to interpret thematic information. KGIS provides facilities for defining a base map, displayed on the graphics screen, which persists from query to query until modified or removed. A base map is defined with the DISPLAY command as follows:

```
  DISPLAY roads, parcels, lakes, streams
```

This command displays a map of all roads, parcels, lakes and streams using the default display formats within the currently defined GEOVIEW. A base map can be modified or reset using the REMOVE command:

```
  REMOVE parcels
      or
  REMOVE *
```

These commands remove just the parcels or the entire base map, respectively. The contents of the base map are recorded in the dynamic legend along with other graphic query results.

Graphical Input

A special graphical input facility is available for specifying a spatial constraint. The user may specify a feature by pointing at a location on the graphic screen with a pointing device, such as a mouse. The feature specified need not be displayed at the time. This facility is employed as follows:

```
  SELECT * FROM parcels WHERE location = mouse;
```

When this query is submitted, the graphic cursor appears on the graphics screen. The user may move the cursor, via the mouse, to a desired location on the screen and indicate a selection by pressing the left mouse button. The user can repeat this until all the selections have been made. The process is terminated by pressing the right mouse button. The query processing then proceeds using the indicated set of features.

CONCLUSIONS

I have presented here the basis of a query language for managing geographic information. The language treats spatial and non-spatial attributes in a single context, providing a high-level relational view of geographic features. Facilities are provided for the graphical display and input of data, the specification of both graphical and geographical context and the use of spatial attributes and relationships as selection constraints. No claims of completeness are made, however. The language currently provides facilities for some of the more common spatial relationships. As facilities for other relationships are added, the language will continue to evolve.

REFERENCES

Abel, D.J. and Smith, J.L.,1985, A RELATIONAL GIS DATABASE ACCOMMODATING INDEPENDENT PARTITIONINGS OF THE REGION, International Symposium on Spatial Data Handling, Seattle, WA, pp 213-224.

Chamberlain, D.D., M.M. Astrahan, K.P. Eswaran, P.P. Griffiths, R.A. Lorie, J.W. Mehl, P. Reisner and B.W. Wade, 1976. "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control", IBM Journal of Research and Development, 20(6), pp. 560-575.

Dueker, K.J., 1985, GEOGRAPHIC INFORMATION SYSTEMS: TOWARD A GEO-RELATIONAL STRUCTURE, Auto-Carto 7 Proceedings, Washington,D.C., pp 172-175

Frank, A.F., 1982, MAPQUERY: DATA BASE QUERY LANGUAGE FOR RETRIEVAL OF GEOMETRIC DATA AND THEIR GRAPHICAL REPRESENTATION, SIGGRAPH Conference, Boston, MA, Computer Graphics Vol. 16, No. 3, p 199.

____________, 1984, REQUIREMENTS FOR DATABASE SYSTEMS SUITABLE TO MANAGE LARGE SPATIAL DATABASES, International Symposium on Spatial Data Handling Proceedings, Zurich, Switzerland

____________, 1986, PANDA: AN OBJECT-ORIENTED PASCAL NETWORK DATABASE MANAGEMENT SYSTEM, Report No. 57, Department of Civil Engineering, University of Maine, 103 Boardman Hall, Orono, Maine 04469

Keating, T., Phillips, W. and Ingram, K.J., in press 1987, AN INTEGERATED TOPOLOGIC DATABASE DESIGN FOR GEOGRAPHIC INFORMATION SYSTEMS, Photogrammetric Engineering and Remote Sensing

Morehouse, S., 1985, ARC/INFO: A GEO-RELATIONAL MODEL FOR SPATIAL INFORMATION, Auto-Carto 7 Proceedings, Washington,D.C., pp 388-397

Peuquet, D.J., 1984, DATA STRUCTURES FOR A KNOWLEDGE-BASED GEOGRAPHIC INFORMATION SYSTEM, Proceedings, First International Symposium on Spatial Data Handling, Zurich, Switzerland, Geographical Institute, University of Zurich, pp 372-391

_____________, 1985, THE USE OF SPATIAL RELATIONSHIPS TO AID DATABASE RETRIEVAL, International Symposium on Spatial Data Handling, Seattle, WA, pp 459-471.

Waugh, T.C. and Healey, R.G., 1985, THE GEOVIEW DESIGN: A RELATIONAL DATABASE APPROACH TO GEOGRAPHICAL DATA HANDLING, International Symposium on Spatial Data Handling, Seattle, WA, pp 193-212

AN INFORMATION SYSTEM FOR GEOSCIENCES: DESIGN CONSIDERATIONS

Marinos Kavouras
Salem E. Masry
Department of Surveying Engineering
University of New Brunswick
Fredericton, N.B., Canada
E3B 5A3

ABSTRACT

The geosciences, and particularly those involved in resource evaluation, are confronted with data handling problems which involve highly complex three-dimensional objects. This complexity demands advanced object representations and data organizations which the conventional GIS is not designed to deal with. The ability to efficiently store, analyze, and update complex geo-information, together with the ability to plan resource explorations, has been the subject of two years of research here at the University of New Brunswick. This paper describes an integrated spatial information system which has been produced as a result of this research.

The system employs various data representations in order to characterize point, line, surface, and solid type geo-objects. These objects can be highly irregular, fragmented, and non-homogeneous. Both geometric shape and the spatial distribution of attributes are known to the system through an extended octree indexing scheme. This organization of the information permits efficient spatial and attribute queries, including boolean operations and queries which involve object topology. Software was written to utilize these abilities within an interactive graphics session.

Date sets were provided by local mining firms in order to assess the suitability and efficiency of the extended octree structures for geo-applications, and the practical utility of the system. Additional applications software was included to perform geostatistical analyses, and operations involved more specifically in the design of a mine. Details and experiences from the practical assessments are also reported in the paper.

1. BACKGROUND

There has been an increasing use of digital technology in the geosciences during recent years. Similar trends have been observed in the areas of resource evaluation and mining but at a slower rate. The main reason for this lag in development rests with the complexity of the problems addressed. Large amounts, and differing types of spatial information compound these problems. The automated collection of digital survey data and use of computerized drafting systems speeds-up laborious work but contributes very little to resource management and efficient mine planning - a complicated task for the geologist and the mine engineer. There is a demand for a geo-information system which efficiently represents, organizes and stores the complex geometry and geology of natural objects so that analyses and updates can be readily performed.

In the mining context, the system should enable the user to store, retrieve, manipulate and display information about:

- the attributes of any point in the mine
- the shape and volume of the entire ore body or parts of it
- the network of excavations and utilities
- the intersection between mining objects such as the ore body and the stopes
- the shortest distance between objects

- the cross sections, profiles and projections of any objects.

In addition, the system should be designed in a flexible and expansible way so that it can accommodate the variety of shapes and attributes involved in all geo-applications. The information-system approach can satisfy the geoscientific needs in a more integrated way than general purpose computer graphics or CAD systems. A number of smaller computer graphics systems developed for certain geo-applications only (e.g. mining), are also very limited in their capabilities because their design has been aimed at the automation of operations which were performed manually in the past. Conceptually, their procedures are still the same, only a lot faster. Such systems solve the specific problems that they were designed for. Expansions are difficult and often impossible, due to simplistic object modeling and data structures.

This paper describes the characteristics of an information system designed for general geoscientific applications. Particular emphasis was given to mining applications as they are very complex. The research and its implementation were done in conjunction with CARIS- a Computer Aided Resource Information System (Masry, 1982; Lee, 1983), which is in use at the Digital Mapping Laboratory of the University of New Brunswick. As a result of this research, a prototype system, known as *Daedalus*, was developed. The most important aspects of the Daedalus design are:

1. The selection of internal representations for the 3-D spatial geo-objects.

2. The way these representations are created, organized and accessed by the system.

The first aspect is the subject of the next section of this paper. Section 3 presents the overall system design with emphasis on the data structure. Results from the practical assessments of the system are reported in section 4.

2. REPRESENTATION OF SPATIAL GEO-OBJECTS

The system developed distinguishes between geometric and attribute geo-information. Geometry relates to the location and shape properties of spatial objects, and consists of metrics and topology. Attributes consist of textual information not related to the shape definition of the objects involved. Geometry and attributes are stored separately.

The system classifies geo-objects into four different geometric types, according to their complexity:

- point type (stations, control points, elevation points),

- line type (utilities, drill-holes, ventilation/power/water lines, transportation, etc.)

- regular solid type (shafts, drifts, ramps, stopes, etc.)

- irregular solid type (most geological objects, stopes, etc.)

The geological information comes primarily from drill-hole data, whilst the input information for the rest of the geo-objects is obtained using standard surveying techniques.

The traditional method of representing 3-D objects is to use multiple orthogonal projections of their surfaces. This method may be sufficient in the case of simple objects, but does not efficiently represent complex objects. In practice, complex objects almost always have to be treated as assemblies of components. Forrest (1978), gives a general complexity measure for objects where:

- *embedding* complexity refers to the dimension of the euclidean space in which the modeled objects are embedded;

- *geometric* (component) complexity refers to the geometry of a component;

- *combinatorial* complexity refers to the number of components in an assembly.

Being irregular and possibly non-homogeneous or fragmented, solid-type geo-objects (geo-solids), such as ore bodies, are the most difficult to model. Man made objects usually have a more regular shape. Utilities (point and line type geo-objects), are easier to handle since, in practice, they are of network type and of constant dimensions. Geo-objects (regular or irregular) present high combinatorial complexity. Therefore, many of the geometric operations to perform involve a large number of components and, as such, give rise to significant computational problems.

Considering the complete representations for solid objects (Requicha, 1980), we are mainly interested in the following two:

- *boundary* representations where the surface which encloses the solid object is modeled

- *volume* representations where the solid's interior is represented as a collection of volume primitives of different size.

In the system developed, a simple and robust volume representation - known as *Linear Octree Encoding*, is used to represent the shape of geo-solids and the spatial distribution of their attributes. The Linear Octree is a hierarchical tree structure proposed by Gargantini (1981). Octrees (Samet, 1984), have found many applications in computer graphics and lately in solid modeling (Meagher, 1982). Very recently, they have also been used to represent topography and geology, (Kavouras, 1985; Mark & Cebrian, 1986). In the octree scheme, the areal extend of the application (such as an entire mine), is enclosed in a large cuboid called "universe". The cuboid universe is subdivided into eight subcuboids (octants) of equal size which are indexed in a specific encoding scheme. If octants contain volume of importance (as in solid modeling), they are called *voxels* (volume elements). Each voxel is attached a color depending on whether it lies inside (BLACK), outside (WHITE), or at the border (GREY) of the geo-solid. The subdivision continues recursively only for the GREY voxels, and terminates when either no GREY voxels remain or when a preset resolution is reached. The smallest elements after the Nth subdivision are called *resolution voxels*. The GREY resolution voxels are of importance because they lie on the surface of the geo-solid.

Due to the spatial coherence of nature, neighboring elements are likely to consist of the same material. By aggregating them into homogeneous regions within an octree structure, and by storing only the BLACK and GREY voxels, substantial storage compression can be achieved. More importantly however, such a compression, in contrast to other techniques (Comeau & Holbaek-Hanssen, 1983), does not suppress the topology of the uncompacted data.

Octree modeling has a number of advantages that make it an attractive scheme for modeling ore bodies in underground or open-pit mines, water reservoirs, caverns, and other geo-solids. Some of the perceived advantages are:

- It creates complete and valid representations.

- It can represent arbitrarily irregular or fragmented geo-solids.

- It stores geometry and basic geology in the same scheme.

- It can also represent the interior of non-homogeneous geo-solids such as ore bodies with variable distribution of grades and other properties. It therefore relates easily to geostatistical block estimations and mine planning.

- Rigid and homogeneous ores can be very concisely represented by exploiting their spatial coherence.

- Its hierarchical nature makes the generalization of geo-solids to variable resolutions very simple.

- Geometric operations useful in geology and mining are easy to perform due to the efficiency of algorithms facilitated by the octree scheme itself.

- Both full and void space can be stored in the same scheme, making volume computations for ventilation analyses trivial.

- Octree modeling can be used in the finite element method, being therefore useful in deformation analysis and rock mechanics, (Chrzanowski et al., 1983).

- The scheme maintains adjacency relations so that different geo-objects (such as ore bodies, excavations and utilities), can be spatially related without extensive searches in an extended data base.

The octree representation scheme has also its disadvantages. It is shift and rotation variant; it is not suitable for surface analysis; and it always involves some approximation when converted to a boundary representation. These disadvantages would be serious in many industrial applications. In geo-applications however, they do not seem to be crucial, and are outweighed by the advantages. For those special geo-applications where both accurate surface description and knowledge of the solid's interior are important, either multiple or hybrid representations (Carlbom et al., 1985; Kavouras, 1986) have to be employed.

In Daedalus, block models are computed from geological sections and polyhedra, (Smart, 1986)(Fig. 1). The system can also utilize block models which have been estimated from a geostatistical package. Octrees are then easily generated by reducing the block models. The procedure for creating representations for ore bodies or other geosolids, can be outlined as follows:

High level programs are used to format and store the geometric and geological core data derived from drill-holes. If a geostatistical block estimation for the entire mine area, already exists, a special octree aggregation/classification procedure (Kavouras, 1986), can be directly used to define the ore body. If this is not the case, a number of interactive steps have to be followed:

-- The geologist retrieves selectively all the drill-hole information which lies within a particular section of certain thickness. He then, semi-interactively defines the ore-waste contact. From correlation of parallel adjacent sections, a complete boundary model of the ore is computed. Next, the boundary model is converted to block data (spatial enumeration arrays), (Smart, 1986).

-- The geological sections can be simple polygons, in which case, no grade variations are distinguished inside the ore. If the ore body is indeed homogeneous, then all blocks carry the same geological attributes. If however, there is an important variation of ore quality, geostatistics can be used to estimate accurate grade values for all generated blocks.

-- The generated blocks are finally converted to the octree voxel representation using the octree aggregation/classification procedure. Both grade values and their estimated accuracy are used to classify ore zones of certain richness.

Whereas modeling of irregular geo-solids is a complex problem, the modeling of regular solids and point/line type utilities is much easier. In Daedalus, the internal representations for regular solids and other utilities consist of the coordinates of characteristic points of the object's axes, and a short list of other geometric (usually cross-sectional) parameters. Namely, the solid is not represented explicitly as in the case of octrees, but only implicitly by some parameters. The actual solid in an octree or other form, may be computed locally when some geometric problem, such as an intersection, has to be solved.

Surface type geo-objects (geo-surfaces), can be treated as thin solids, and be discretely represented as octrees. Some applications however, may require an explicit surface description. The data structure of Daedalus has been designed to accommodate simplicial composite surfaces with planar polygonal faces, and full topological surface-face-vertex description (Baumgart, 1975). Complex handling however is still under development.

3. SYSTEM DESIGN - DATA STRUCTURE

In order to explain the different modules of the system, consider the logical system design, as shown in figure 2a.

At the lowest level of the system stands the data base. It consists of the digital shape representations of all objects, their attributes, and nothing else.

At the next higher level - the data structure of the system, there exist a number of basic facilities for organizing new information in the data base or for accessing previously stored data. This organization also ensures the integrity of the information (such as validity of representations). Spatial searches in given locations and their neighborhood are facilitated here. Elementary but robust operations on data base entities, such as octree aggregation of neighboring voxels, also belong to this level.

The next level contains modeling and low level general operations. Here internal representations of geo-objects are created and/or converted. Geometric operations perform volume/surface computations, projections, sections, and transformations on the stored objects. Set (boolean) operations perform union, subtraction or intersection on octree encoded objects (Reeler, 1986).

At the highest level, there are operations to answer complex metric, topological and attribute queries. Other algorithms perform hidden surface removal on selectively displayed mine sections. Finally, application programs can perform user requested operations, such as contouring, volume computations, geostatistical evaluations, interference/adjacency analysis, and so on.

In order to provide the necessary efficiency, the data structure of Daedalus consists of five sub-structures, (Fig. 2b):

-- The VOXEL structure which contains data files and access methods to the octree representations of all geo-solids. Voxel geometry and basic geology can be stored together, whilst additional attributes are kept in the attribute files. The file structure allows for direct accessing of single voxels and their attributes. The so far experience with geo-applications shows that direct accessing to single pieces of information is essential and always requested by the users.

-- The SURFACE structure which contains data files and access methods to the simplicial composite surface representations of all geo-surfaces. Integration of this structure into the Daedalus system is still in progress.

-- The CARIS data files and access methods to all representations of point, line, and regular solid type geo-objects. The CARIS structure can handle high densities of utility data with very satisfactory performance.

-- The CUBEL space subdivision indexing scheme which points to all point, line, surface or regular solid type objects of each specific CUBoid ELement in the application universe. The cubel size is based on the density of spatial information. Global density criteria result to a fixed cubel size. Local density criteria require a dynamic and variable cubel size which has to be updated as information is added or deleted from the data base. Empty

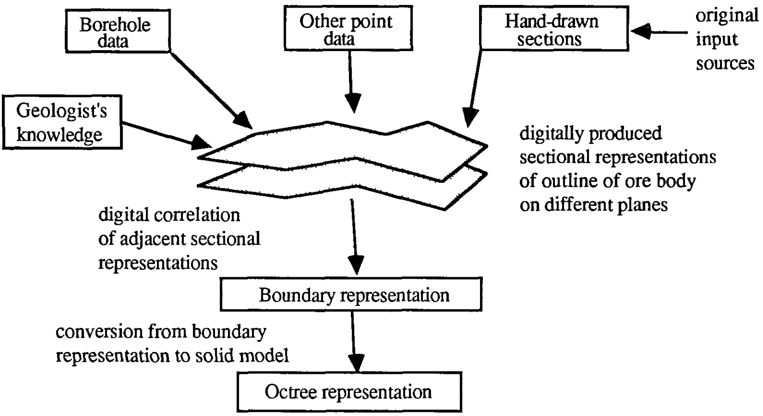

Borehole data
Other point data
Hand-drawn sections
original input sources
Geologist's knowledge
digitally produced sectional representations of outline of ore body on different planes
digital correlation of adjacent sectional representations
Boundary representation
conversion from boundary representation to solid model
Octree representation

Figure 1 : From Sectional to Octree Representations, (from Smart, 1986).

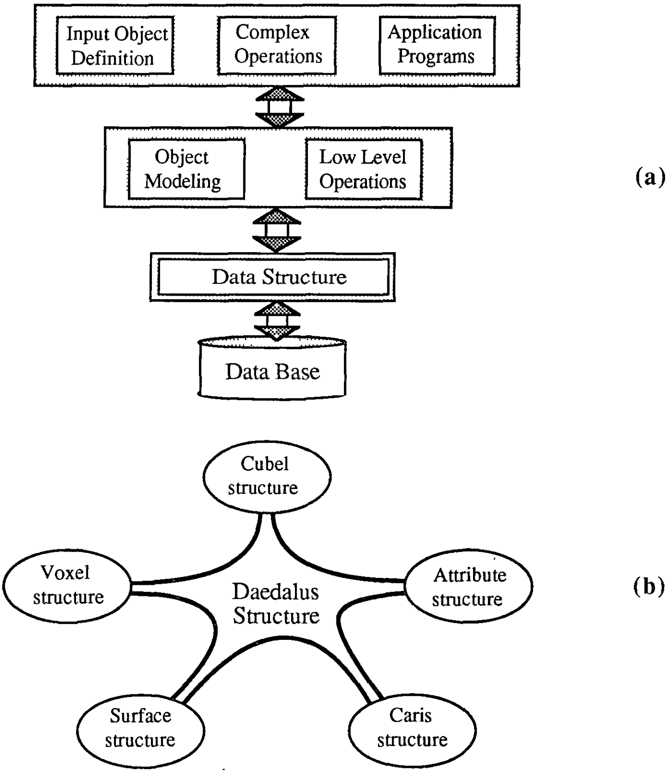


Figure 2: System Design: Daedalus Modules (a) and Structure (b).

cubels are not stored. By basing it on a similar octree-encoding-scheme, the cubel structure is implicitly linked to the voxel structure through the use of linear keys. Results from most applications with a fixed cubel size have been satisfactory. More analysis is however still required in order to determine the optimal cubel size for variable densities and operations.

-- The ATTRIBUTE structure, which contains textual information not directly related to the geometry of the spatial objects. Any attribute information, such as utility maintenance records, land ownerships, detailed geological descriptions, and exploitation history are stored in attribute files and are accessed via a commercially available DBMS.

4. PRACTICAL ASSESSMENTS

The system has been assessed using data sets provided by local mining firms. The assessment was based on various mandatory requirements for a suitable spatial information system. A summary of those requirements were:

- Suitability of the selected schemes in representing point, line, and regular-solid type geo-objects.

- Suitability of the octree scheme as a representation of irregular geo-solids, and particularly ore bodies with variable distribution of ore quality.

- Sufficient compaction for homogeneous and non-fragmented ore bodies.

- Fast access to any piece of spatial information, without extensive searches.

- Efficient interactive geometric manipulations on block models which involve up to hundreds of thousands of blocks. In particular, boolean operations, sectioning, and volume computations are very essential to mine design and planning.

- Integration with all mining objects and surface topography.

- Attribute analysis based on user specified criteria.

We present below a real mine test case of an underground gold ore deposit. The block grades and their accuracy had been previously estimated using geostatistics. The block model consisted of cubic blocks (20x20x20) cubic feet, and the areal extend of the estimation was (3800 x 1600) square feet, with a vertical thickness of 840 feet. Therefore, the model consisted of (190x80x42) = 638,400 single blocks. The range of block grades varied (non-uniformly), between 0.000 and 5.368 ounces of gold per ton.

Being quite fragmented, the ore body could not be modeled by a boundary representation, and the octree scheme was ideal to use as an internal representation. Using a cut-off grade of 0.100 ounces/ton, the blocks were classified as waste (below cut-off grade), or ore (above cut-off grade). The resulted ore body consisted of only 30,412 blocks. A subsequent octree aggregation based only on adjacency, resulted in an ore body of 17,007 voxels of different sizes (Fig. 3). The ore grades varied between 0.104 and 5.368 ounces/ton, with a mean single block grade of 0.504 ounces/ton, and a standard deviation of 0.286 ounces/ton. Since the number of voxels is directly proportional to the surface area of the encoded solid (Meagher, 1982), the compaction would have been much higher if the ore body was not so fragmented.

In order to assess effectiveness and efficiency of geometric operations, excavations were then designed in the form of shafts, ramps, drifts and stopes, for three major levels (Fig. 4). The surface topography was also encoded in the data base (Fig. 4). Intersections were then computed among excavations, the ore body, and the drill-holes. In this way, any undesirable interference was avoided, and the stopes were optimally positioned with respect to the ore body (Fig. 5). Volumes of recoverable ore and of void space were then

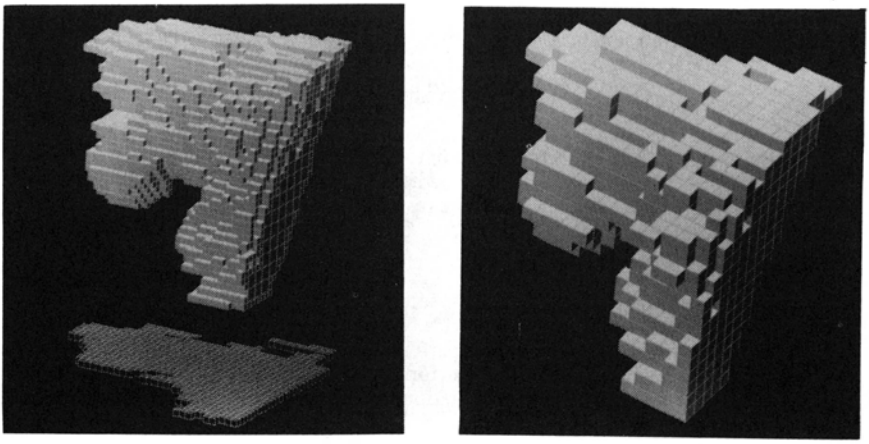


Figure 3 : A Portion of the Ore Body, Detailed and Generalized.

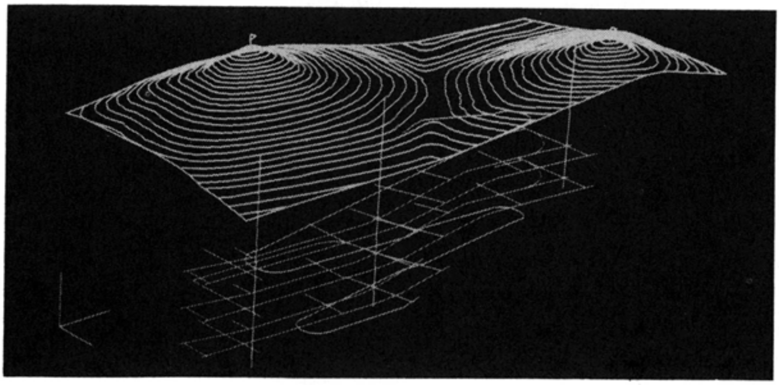


Figure 4 : Surface and Portion of Underground Utilities.

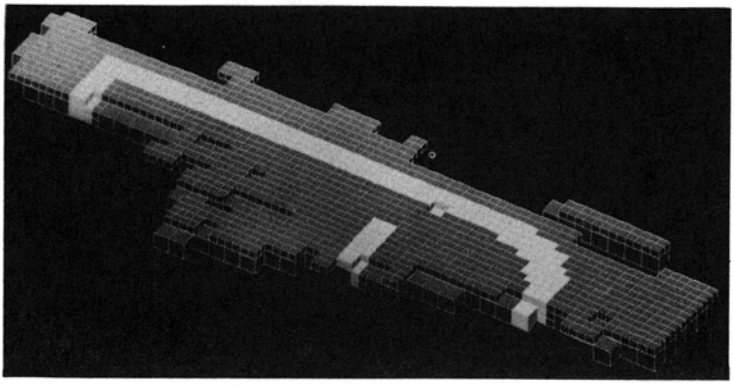


Figure 5 : Embedded Excavation in the Ore Body.

computed. The first volume computation serves the production planning, and the second serves the ventilation analysis. The above operations were performed in the area of interest each time (e.g. stopes and ore of one level at a time), without involving other parts of the data base. This significantly increased the efficiency of the operations.

Attribute analysis was also assessed, in which queries such as: "Retrieve all blocks which are of certain size and have an ore grade between z1 and z2 ounces/ton", were invoked. Such a query generates an attribute report where a summary of all blocks found is listed, along with their position with respect to the rest of the ore. Volumes and statistics are also displayed on a graphics screen or printed. Other queries, such as: "Find the shortest distance between certain drifts/stopes and the rich blocks of the ore", were also assessed.

The overall assessment showed that the selected schemes and particularly the octree encoding, are suitable to represent a wide variety of mining geo-objects. Geometric operations - essential to mine design, are performed efficiently. Attribute analyses are also satisfactory. All these queries can be performed comfortably in an interactive session. There are however, a number of optimizations which are required, in order to make the prototype system a production system:

- Daedalus structure has to be fully integrated with the CARIS structure and peripheral utility programs.

- Additional tests and assessments with other mining conditions, or geo-applications are also needed.

- The main system should include some additional peripheral utility programs, such as 3-D network analysis for route selection in mine planning, some aided-design functions, and more sophisticated display programs. Integration with a digital terrain modeling and analysis package also appears to be desirable.

5. FINAL REMARKS

A prototype Spatial Information System for mining applications has been developed and tested at the University of New Brunswick. Its sophisticated design proves to be flexible and expansible in its handling of geo-applications. Development will still continue in order to optimize certain system modules, and add some peripheral utilities. Results have been very satisfactory and local mining firms have expressed a strong interest in adopting a production system.

In order to satisfy as many adverse applications as possible, future research will be directed towards the incorporation of complete surface representations in the Daedalus system. Also, the complexity, usefulness, and potential implementation of hybrid and multiple representations will be investigated.

6. ACKNOWLEDGEMENTS

This project has been funded by the Natural Sciences and Engineering Research Council of Canada; this support is greatly appreciated. Many organizations and companies have also contributed to the reported work with data and information. We wish to thank in particular: The Canada Centre for Mineral and Energy Technology-EMR, Ottawa; Universal Systems, Ltd., Fredericton, N.B. Canada; The New Brunswick Department of Forests, Mines & Energy - Mineral Resources Division, Fredericton, N.B., Canada; The Potash Company of America, Sussex, N.B., Canada; The Brunswick Mining and Smelting Corporation, Ltd. - Mining Division, Bathurst, N.B., Canada; and Voest-Alpine AG, Eisenherz-Leoben, Austria. We are also grateful to numerous individuals for their constructive comments during many fruitful discussions.

REFERENCES

Baumgart, B.G. (1975) - "A Polyhedron Representation for Computer Vision", *AFIPS Conf. Proc., NCC,* Vol. 44, pp. 589-596.

Carlbom, I; I. Chakravarty; D. Vanderschel (1985) - "A Hierarchical Data Structure for Representing the Spatial Decomposition of 3-D Objects", *IEEE CG&A,* April, pp. 24-31.

Chrzanowski, A,; Y.Q. Chen; A.S. Chrzanowski (1983) - "Use of the Finite Element Method in the Design and Analysis of Deformation Measurements", *Proc. of XVII FIG Congress,* Commission 6, 611.1, Sofia.

Comeau, M.A.; E. Holbaek-Hanssen (1983) - "Compression and Compaction of Binary Images", *Proc. of AUTO-CARTO VI,* Vol. I, pp. 362-371, Ottawa.

Forrest, A.R. (1978) - "A Unified Approach to Geometric Modeling", *ACM Siggraph Computer Graphics* 12, 3, pp. 264-269.

Gargantini, I. (1982) - "Linear Octtrees for Fast Processing of Three-Dimensional Objects", *Comp. Graph. & Image Proc.* 20, pp. 365-374.

Kavouras, M. (1985) - "Design of a Geometry-System to Handle 3-D Mining Information", *Proc. of VI Inter. Congress on Mining Surveying,* Harrogate, England, September 1985.

Kavouras, M. (1986) - *PhD Thesis in preparation,* Dept. of Surveying Engineering, Univ. of New Brunswick, Fredericton, N.B., Canada.

Lee, Y.C. (1983) - "A Data Structure for Resource Mapping with CARIS", *Proc. of AUTO-CARTO VI,* Vol. I, pp. 151-160, Ottawa.

Mark, D.M.; J.A. Cebrian (1986) - "Octtrees: A Useful Data Structure for the Processing of Topographic and Sub-surface Data", *Proc. of ACSM-ASPRS Annual Convention,* Vol. 1, pp. 104-113, Washington, D.C., March 1986.

Masry, S.E. (1982) - "CARIS - A Computer Aided Resource Information System: An Overview", *paper pres. at the Institute for Modernization of Land Data Systems,* Georgetown University Law Centre, Washington, D.C., January (rev. Sept. 1982).

Meagher, D.J.R. (1982) - "The Octree Encoding Method for Efficient Solid Modeling", *IPL-TR-032,* Image Processing Lab., Rensselaer Polytechnic Institute, Troy, N.Y.

Reeler, E.C. (1986) - "The Manipulation of Linear Octrees in a Three-Dimensional Digital Mapping System", *MScE Thesis,* Dept. of Surveying Engineering, Univ. of New Brunswick, Fredericton, N.B., Canada

Requicha, A.A.G. (1980) - "Representation of rigid solids: theory, methods and systems", *ACM Comp. Surveys* 12, pp. 437-464.

Samet, H. (1984) - "The Quadtree and Related Hierarchical Data Structures", *ACM Comp. Surveys,* 16, No. 2 (June), pp. 187-260.

Smart, J.R. (1986) - "Three-Dimensional Modelling of Irregular Natural Objects", *MScE Thesis,* Dept. of Surveying Engineering, Univ. of New Brunswick, Fredericton, N.B.,Canada.

ATTRIBUTE HANDLING
FOR
GEOGRAPHIC INFORMATION SYSTEMS

Peter Aronson
Environmental Systems Research Institute
380 New York Street
Redlands, CA 92373

ABSTRACT

Geographic information systems manipulate and manage both spatial information and the thematic attributes of that information. There are several candidate methodologies for the management of these thematic attributes for system designer to choose among. Which is the most useful, both in terms of data model and of normal usage? This paper discusses the choices open to the system designer in the context of both sets of criteria.

1 INTRODUCTION

Real world geographic entities can be modeled in a Geographic Information System· (GIS) as features composed of a set of locational information (position, geometry and topology) and a set of thematic information. The handling of locational information is beginning to be somewhat understood, and there exist widely accepted paradigms to deal with it (such as the topological model). The handling of thematic data, of sets of attributes, while well understood in general, is not well understood in terms of GIS processing.

The subject of thematic data handling has been very well studied in general however. Database Management Systems (DBMS) have been in use for well over twenty years, and in that time many advances have been made. Modern DBMS manage data using sophisticated techniques drawn from various branches of mathematics (such as set theory and graph theory) as well as the latest techniques of computer science. Several of these techniques have been incorporated into existing GIS.

The approach used to manage thematic data can not be examined independently of the GIS data processing model upon which it is based. A GIS a is geographic database and a set of operations upon that database - the form of operations performed on the spatial portion of that database specifies the form of operations required upon the thematic portion.

While the data models and techniques used to manipulate thematic data are important, equally or more important are the organizational procedures involved in that data's collection, evaluation and use. Organizations that produce and use data have needs distinct and separate from the requirements of the software. The organization is not going to change, so a GIS's thematic data handling must be able to match that organization's needs, or it will not be used.

This paper is organized into five sections: an introduction into the nature of the problems involved in thematic data handling for GIS; a survey of thematic data models currently in use; a discussion of GIS processing models and their implications for associated thematic data processing; a discussion of the organizational constraints on the handling of thematic data; and finally, conclusions on the above.

2 THEMATIC DATA MODELS

There are many different paradigms for the management of thematic data. The most common are: Tabular; Hierarchical; Network; Relational; and Object-Oriented. The first is the manner in which most early GIS stored their attribute data (if any), the next three are those currently most commonly implemented in DBMS, while the last is newer but rapidly gaining in popularity for some applications.

The simple tabular model sees data as collections of independent tables with rows (records) and columns (fields). These usually will have fixed field definitions, but aren't required to. Fields may be variable length or repeating. Such systems will usually have simple query systems if at all.

2.1 Simple Tabular Model

The simple tabular model allows the association of attribute codes with geographic features. Its major lacks are in terms of data integrity (since each table is independent, identical data to be used with two different tables must be present in both, which means they can disagree), storage efficiency, and flexibility; however such data structures are easy to program and to convert from system to system.

2.2 Hierarchical Model

A hierarchical database organizes data in a tree structure. A tree is composed of a hierarchy of elements. The uppermost level of the hierarchy has only one element, the root. With the exception of this root, every element has one element related to it at a higher level, referred to as its parent. No element can have more than one parent. Each element can have one or more other elements related to it at a lower level, referred to as that element's children (Martin, 1975).

Hierarchical DBMS have not gained any noticeble acceptance for use in GIS. They are oriented for data sets that are very stable, where primary relationships among the data change infrequently or never at all, since the data relationships are built into the logical view of the database. Also, the limitation on the number of parents that a element may have is not always found in actual geographic data (the section of US Highway 215 immediately south of US 10 would have two parents in the California Highway database, US 215 and California 91). Finally, the query language for a hierarchical DBMS is of necessity procedural, that is, it requires knowledge by the user of the actual storage scheme used by the DBMS. This is information that the user should definitely not be required to know.

2.3 Network Model

A network database organizes data in a network or plex structure. Any item in a plex structure can be linked to any other. Like a tree stucture, a plex structure can be described in terms of children and parents. In a plex structure, children may have more than one parent, and link back upwards (that is, an element can be its own grandparent or even parent) (Martin, 1975).

Network DBMS have not found much more acceptance in GIS than hierarchical DBMS, although they are not without their champions (Frank, 1982). They have the same flexibility limitations as hierarchial databases; however, the more powerful structure for representing data relationships allows a better modelling of the data relationships found in geographic data. The query language, however, for network databases is still procedural.

2.4 Relational Model

In a relational database, information is organized in tables. These tables have a more rigorous definition that in the simple tabular model. The tables, which are identified by unique table names, are organized by column and row. Each column within a table has a unique name. The set of values from which the actual values in a column are drawn is called the domain of the column, and may be shared among different columns (within different tables). Each row (or tuple) is a set of permanantly associated values. Tables may be joined to each other by columns sharing a common domain. Such joins are usually ad hoc and temporary operations, unlike the previously discussed database types, these relationships are implicit in the character of the data as opposed to explicit characteristics of the database set up. A simple example of a join of two tables in a relational database:

SOIL_POLYGONS

Poly#	Area	Soil_Code
1	37.5	AN 32
2	15.6	CE 12
3	41.7	BG 17
4	22.1	AN 32

SOIL_DATA

Soil_Code	PH	Sample_Date
BG 17	-1.7	11/23/84
CE 12	+3.2	04/06/82
AN 32	+1.7	12/22/81

Since both Soil_Code columns share the same domain (legal soil type identifiers), the two tables can be joined by soil code. This yields the resulting table:

SOIL_POLYGONS + SOIL_DATA

Poly#	Area	Soil_Code	PH	Sample_Date
1	37.5	AN 32	+1.7	12/22/81
2	15.6	CE 12	+3.2	04/06/82
3	41.7	BG 17	-1.7	11/23/84
4	22.1	AN 32	+1.7	12/22/81

Note this result need not be an actual table, but can be generated as required. This results in a smaller storage requirement (there is no redundent storage of information for soil AN 32), and a more normalized data structure (see below). Note, a different result could be produced by joining the Soil_Polygon table with yet another table, say a Polygon_Symbology table, joined by the Poly# domain.

The relational database model is the most widely accepted for managing the attributes of geographic data, examples including SGIS, GEOVIEW (Waugh & Healy, 1986) and, ARC/INFO (Morehouse, 1985). It is attractive because of its simplicity (all data stored in simple tables), its flexiblity (any set of tables can be joined together by columns with common domains), efficiency of storage (by proper design of tables, redundant information can be eliminated) and by its non-procedural nature (queries on a relational database do not need to take into account the internal organization of the data). The relational DBMS has emerged as the dominant commercial data management tool of the eighties.

2.5 Object-Oriented Model

The basic unit that an object-oriented DBMS manages is the object. An object is a collection of data elements and operations that together are considered a single entity. Objects are typed, and the format and operations of an object instance are the same as some object prototype. Objects may be hierarchical, that is, objects may be composed of other objects in turn (Wiederhold, 1986). An example of a object might be a swamp object:

Swamp Object:

List of Border Chains: C1, C2, C3,...,Cn

List of Nodes: N1, N2, N3,...,Nn

Attributes: Depth
Muck type

Soil Samples: S1,...,Sn

Symbology: Solid borders
blue shade
random swamp symbols

Operations: Measure
Drain
Expand

Once this structure is set up, the details of it need not be user visible. The above is a relatively passive view of an item. In some systems objects take a very dynamic role, being the primary means for rules to be implemented.

As noted above, the object-oriented database is a relatively new model, although its origins go back to work done at Xerox in the early seventies. So far, the only geographic data handling system to extensively employ this model is TIGRIS (Wientzen, 1986). This approach has the attraction that query is very natural, as features can be bundled together with attributes at the database administrator's discretion. It is however considerably less ad hoc than the relational model, and is not normalized.

In addition to the above pure systems composite systems exist as well that combine characteristics of two or more models, such as relational-hierarchical or object-oriented-relational.

3 GIS PROCESSING MODELS

In general, the form of thematic attribute processing appropriate for a GIS depends on the data processing model that it uses. A data processing model is a formalization of operations on data, as opposed to a data representation model, which is a formalization of a real world object or structure (an example of a GIS data representation model is the USGS DLG format, which is a formal model of a USGS topographic quad sheet).

In the context of this discussion, map processing will be discussed independent of the data structures and algorithms involved. In these terms polygon overlay and grid cell overlay are the same operation - spatial join. Only the operations are considerered, not the algorithms nor the representation of the maps themselves. There are three such models commonly used for mapping: the simple map; the composite map; and the relational map.

3.1 Simple Map Processing Model

The simple map processing model assumes that a data set represents a single map sheet. Each data set is thematically atomic, that is, it can not be split into multiple maps by subject - there is only one or no sets of attributes per data set. All attributes are tightly bound to the map; there is no thematic data available except that one set. And it is thematically independent - data sets can not be combined. Examples of simple map models are CAD/CAM systems or many of the simple mapping packages (such as SYMAP).

The simple map model, if it has any thematic data handling at all, uses the simple tabular approach or something functionally equivalent to it, since there is no access to other thematic data or map data sets. There is no need for systems that can handle combined or linked data sets as they never occur.

A pure contouring package would be an example of the simple map processing model. All operations occur upon one set of data (points) and involve only that set of data and its attributes (elevations). There exists no mechanism for joining two data sets by spatial domain (locations) and all operations involve only one data set at a time. The operations are all in the form of F(ds1) -> ds2 (where F is a function on a data set such as rescaling, and ds1 and ds2 are data sets) or the form F(ds1) -> Vds1 (where Vds1 is a "virtual" data set on the order of Moellering's virtual map (Moellering, 1984), and F is an function on the data set such as contouring where the output is a graphic or report). Of neccessity, the types of operations on the thematic data is limited to the types of operations on data sets as a whole.

3.2 The Composite Map Processing Model

The composite map processing model assumes that a data set is a combined set of map sheets. If you add to the simple map model the operation of spatial joining, of overlay, the result is the simplest form of the composite map processing model. In the composite map model, because spatial joins can have occurred, there will be N sets of attributes for each data set, where N is the number of source map sheets that contributed to the data set. The thematic data available for a data set is the sum of all the original map sheets. Operations using the composite map model occur within an assembled data set - combination occurs before further processing.

Attribute handling for this composite processing model can take one of two basic approaches. Once again the simple tabular model can be applied, in which case during the construction of the composite data set, a composite attribute table is also constructed. Attribute operations then occur on this table. Operations occur on the resulting composite spatial elements. The alternative approach is to classify the results of the combination into objects. These objects reassemble the original pre-combination features out of the smaller post-combination elements. These objects contain as a result all the combinations of data overlayed by the resulting object as it is pointed to by the post-combination elements. This allows data to aggregated as needed.

An example of the composite map processing model is a simple GIS with overlay capabilty such as GRID (Tomlinson, et al, 1976). Operations still occur on only one set of data, with the exception of one particular operation, the overlay. After an overlay (an operation of the form F(ds1,ds2) -> ds3) there is a thematic data set with combining the thematic data from all input data sets, permanantly joined by common spatial domain, that is, by the common resulting grid cell. Query and reporting operations can now operate on this composite data set, performing such operations as identifying cells that have value A in ds1 and value B in ds2. A more sophisticated system might be able to identify features that are borders between value A is ds1 and B and in ds2 and have value C is ds3. In all of these cases, operations can only happen on data that has been built into the composite data set.

3.3 The Relational Map Processing Model

The relational map processing model looks at a data set as a set of spatially overlapping, independent map sheets and associated attribute tables. These map sheets are combinable but not permanantly combined. Each map sheet represents a normalized relation consisting of a spatial key (location) and a set of attribute tables. Operations within the relational processing model occur ad hoc as needed between independent elements of the the data set. Also, unlike the above two models, the data set is not sharply bounded - any available data in the proper format may be included in an operation with any other data (assuming they share either a spatial or a thematic domain).

The obvious data management model for the relational map processing model is relational, since it is essentially an extension of the relational model by the addition of spatial joins (overlays). That is, both deal with data sets that can be joined on common domains as required. A useful extention to this model is to allow these joins to occur across multiple DBMSs.

Within this data model, ideally each attribute table, whether attached to a map sheet or not, should be in at least third normal form (3NF). A table is in 3NF if every determinant is a candidate key (Date, 1975). A determinant is an attribute upon which another attribute is functionally dependent, such as PH is functionally dependent upon Soil_Code in the SOIL_DATA table above. A candidate key is a column or a set of columns whose attribute values uniquely identify all the rows in the table. Even more desirable is a further degree of normalization, fourth normal form (4NF) which requires a further degree of independence. What this means in functional terms is that all of the data in a single table should deal with different aspects of a single subject. This is very important for updating that data (see below for discussion).

A partial example of the relational map processing model is the ARC/INFO GIS (Morehouse, 1985). While operations can occur in single data sets as in the simple map processing model or in combined data sets as in the composite map processing model, there is a third fashion in which operations can occur in the relational map model. That is across two or more independent data sets. In theory this operates much like operations in a relational data base. The user specifies a series of spatial and thematic joins and subsetting objections to create a virtual data set (called a view in a relational DBMS), then operates on this virtual data set as if it was physically existent. The virtual data set would never exist as an actual data set. In practice, spatial joins are difficult enough and costly enough so that they are not practical to perform in an ad hoc manner. The technique used in ARC/INFO for relational map processing is to perform overlays on data sets containing no direct thematic data, but simply pointers to other tables containing it. The data sets resulting from this operation act as indices to allow relations between separate data sets.

An example of this would be to take three map data sets; a soils map, a land use map, and a vegetation map; and four associate thematic tables; soils data, land use data, lease data, and vegetation data. Relations are as follows: soil map -> soil data, land use map -> land use data -> lease data, and vegetation map -> vegetation data. The three maps data sets would be overlayed, producing a map data set that had pointers to three thematic data sets (soils, land use and vegetation). The relational database would then be used to link these five data sets (the four thematic data sets and the index data set) together to answer such queries as "Find those polygons that have arable land, no protected species, and are owned by the state".

It should be noted, that as in the thematic data models, the GIS processing models can also exist in hybrid form. There exist GIS that essentially employ the composite map processing model, but have limited relational capability. To even further confuse attempts at classification it is possible to use a system that employs the composite map processing model as if it used the simple map processing model, or to employ a system that uses the relational map processing model as if it used the composite map processing model. Sophisticated capabilities tend to be ignored by users who don't need them.

4 ORGANIZATIONAL CONSIDERATIONS

Organizations acquire geographic information systems to meet their needs - not the other way around . To be successful a GIS must be able to support the organization's existing internal structure. Attempts to change this will typically run into massive bureaucratic inertia, particularly if the current structure is functioning in a satisfactory fashion.

In most government organizations involved in using public land records, as well as in large corporations that collect map data for their own use (such as oil companies), not all the map data for a region is collected by the same agency. In fact, typically, map data will be collected and maintained by a combination of Federal, State and Local agencies. In the Dane County, Wisconson example described by Chrisman (Chrisman and Niemann, 1985), the seven layers in the database were provided by five organization, two federal, one state and two county. This is typical of land records information in this country. (In the commercial sector the situation can be even more complicated since data is often purchased from multiple service bureaus.)

To make matters more complicated, most agencies usually will have begun automation of their thematic data well before obtaining a GIS. This means that the data will be stored in some DBMS system or another. Often conversion to the GISs own format is undesirable or impractical (such as when the DBMS has capabilities that the thematic data handler for the GIS lacks, such as concurrent access control or a powerful report generator). This situation often leads to the requirement that the system handle thematic data in multiple DBMS.

Not only is data typically provided by an assortment of agencies in a variety of forms, it will usually be maintained by the providing agencies. That data will need to be updated, often frequently. It is here that a normalized database pays off. In a properly normalized database, the data sets (tables and maps) are kept divided into elements that only contain data on one subject, and hence, only from one source agency. Since these data sets are not combined until required, each agency can update its data when needed, without worrying about modifying another agency's data. A virtual data set generated at a later time would then automatically incorporate the latest data. This can be particularly important with certain types of thematic data that are updated so frequently as to require transactional capability in the DBMS that stores it, such as statewide land ownership.

5 CONCLUSIONS

The GIS implementor (by which is meant either someone designing a new GIS or someone integrating an existing system into an organization's operations) has not only the task of modeling some portion of the real world for an organization, but of doing so in a manner supportive of the organization's internal structure. Since data is not typcially collected or even processed by a single, centralized source, this requires the processing of thematic data as independent data sets that can be combined as needed. The primary existing tool for this task is the relational DBMS, and the most practical environment in which to apply it is in a GIS that implements the relational map processing model. Current and future GIS systems would do well to work towards this goal.

REFERENCES

Chrisman N. and Niemann, B., Alternative Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning, Proc. AutoCarto 7, 1985. p. 84-93.

Martin, J. 1975, Computer Data-Base Organization, Prentice-Hall, New Jersey.

Moellering, H., Real maps, virtual maps, and interactive cartography: Spatial Statistics and Models, G. Gaile and C. Willmott (eds).: Boston, Mass, D. Reidel, 1984. p. 109-132.

Morehouse, S., ARC/INFO: A Geo-Relational Model for Spatial Information: Proc. AutoCarto 7, 1985.p. 388-397.

Tomlinson, R., Calkins, H. and Marble, D., Computer Handling of Geographic Data: The UNESCO Press, 1975.

Waugh, T. and Healy, R., The GEOVIEW Design: A Relational Database Appraoch to Geographic Data Handling: Proc. 2nd Intl. Symposium on Spatial Data Handling, 1986. p. 193-212.

Wiederhold, G., Views, Objects, and Databases: Computer, Dec. 1986, p. 37-44.

Wientzen, B., TIGRIS.. An Object-Oriented Approach to Topology: InterVue, 4th Qtr., 1986.

A GEOGRAPHICAL DATABASE SYSTEM

Willem van Biljon
National Research Institute for Mathematical Sciences
CSIR, P O Box 395, PRETORIA, 0001, South Africa

ABSTRACT

A geographical database that forms the kernel of a geographical information system (GIS) is described. The database facilitates topologically structured spatial attributes as well as a poset (partially ordered set) classification and non-spatial attributes scheme. The database provides the capability of attaching more than one set of spatial attributes to a single feature to enable the GIS to perform semi-automatic generalization.

INTRODUCTION

The National Research Institute for Mathematical Sciences (NRIMS) of the Council for Scientific and Industrial Research (CSIR) undertook research into geographical information systems (GIS) by developing a prototype computer-assisted cartography system. The development and use of this system has led to a better understanding of the issues involved, and NRIMS is currently designing and implementing a complete GIS.

The major requirements for the GIS are that, in addition to the cartographic capabilities, it should provide fully interactive query facilities, both graphically and alphanumerically. This paper describes the design of the geographical database that forms the kernel of the GIS and in which all feature attributes are stored. Both the objectives for and the resulting design of the database are discussed.

OBJECTIVES

A number of objectives were set concerning the management of attributes in the database. These are the result of two factors. Firstly, a number of requirements were identified during the use of a prototype cartographic system, and secondly, some practical objectives were set in accordance with theoretical objectives of the system. Of importance in this regard are those requirements that relate to completeness of the database. That is, its capability to answer any metric, topological or geographical query. These requirements have been examined by (White, 1984). A summary of the objectives follows.

Data integrity, consistency and reduced redundance

These requirements are usually found in any database design.

Maintenance of data topology

To enable the database to contain enough information to answer all topological and geographical queries it is essential that the

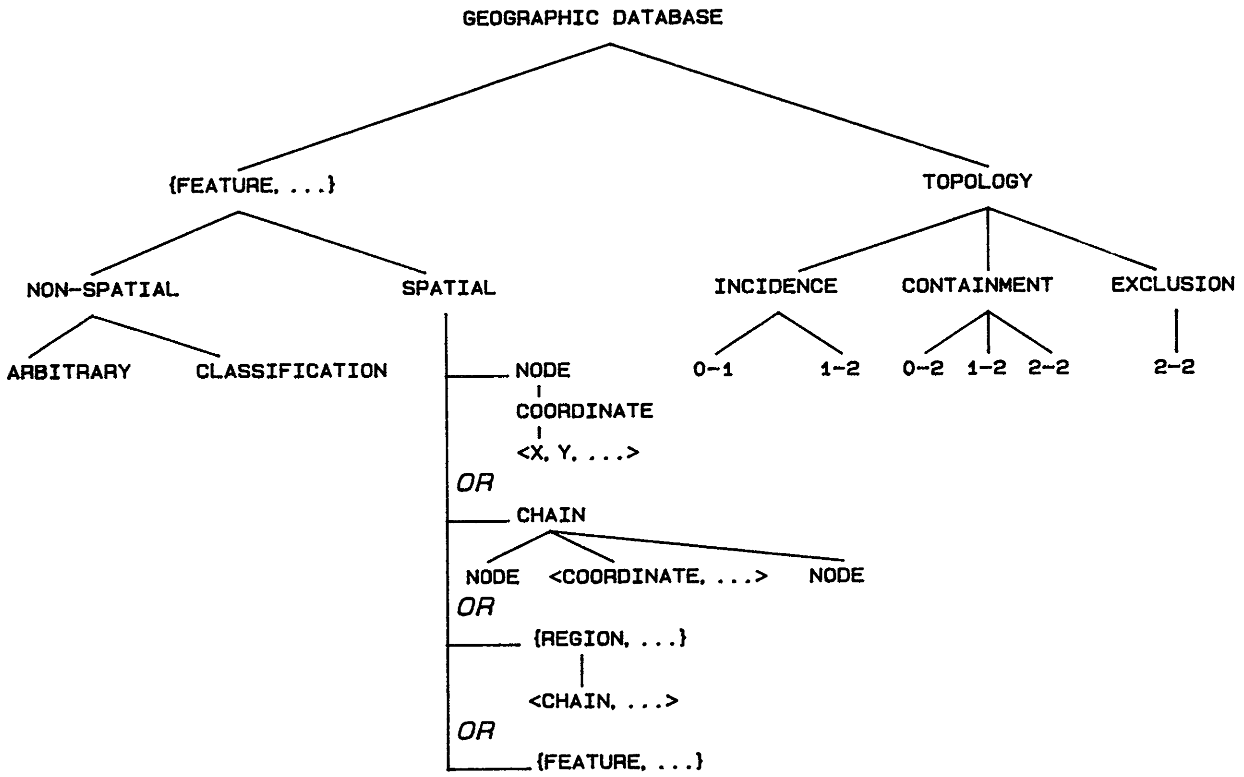


Figure 1: Structure of the Database

topology of the data should be maintained. This information should be independent of spatial attributes since the topology is a function of structure rather than spatial position.

Efficiency of spatial manipulation

Large volumes of spatial information are stored in a GIS. It is essential that efficient data structures should be provided to allow fast interactive manipulation and display of information.

Multiple spatial data sets per feature

Provision must be made to associate a number of different spatial data sets for a single feature. This will allow the digitization of different views of a feature, and may be used to associate data sets that have varying amounts of detail with a feature.

Decomposition of features into sets

It should be possible to construct features from sets of other features. This has two benefits. Firstly, it allows sets of features to be associated, and hence to have a semantic connection. Secondly, this facility, together with multiple data sets, allows the system to perform generalization.

Management of non-spatial attributes in a feature-dependent manner

Features of different types (or classes) have different attributes. A mechanism must be provided to manage these attributes in a feature-dependent manner.

The database was designed with these objectives as primary concerns.

CONSTRUCTION OF THE DATABASE

Figure 1 gives an informal graphical overview of the design of the database. The tree structure depicts each entity in the database, as well as the components of which it consists. Curly brackets denote sets of entities, whereas angle brackets denote arrays of entities, that is, order is important.

The database consists of two components: a set of features and a collection of topologies on the spatial attributes of the features.

Features

A feature represents a natural, man-made or abstract object that exists on the surface of the earth. A feature is described by two types of attributes: spatial and non-spatial. Each feature has an array (or vector) of spatial attributes associated with it, as well as a single set of non-spatial attributes.

Spatial attributes. Each spatial attribute of a given feature is either a point attribute (and we speak of a **point feature**), a line attribute (**line feature**) or an area attribute (**area feature**). Finally, the spatial attribute may be a set of other features (and we speak of a **compound feature**).

A point feature maps directly down to a node, which in turn consists of a coordinate in n-space, usually either on the projection plane or on the spheroid.

A line feature maps down to a list of chains. A chain is a list of coordinates terminated at both ends by nodes.
An area feature maps to a set of regions. Each region is enclosed by a closed boundary constructed from a list of chains.

The nodes terminating a chain may be shared by different chains and point features. Likewise, chains may be shared by different line features and region boundaries. Finally, area features may share different regions. This data sharing addresses the objective of reduced data redundance, and simplifies consistency checking in the database.

Multiple spatial attributes for a single feature are based on the philosophy that spatial attributes may differ according to use; these attributes do not necessarily attempt to reflect a single model of reality. The structure within each element of the vector of spatial attributes is similar to structures described in the literature (Guptill, 1986 and Peuquet, 1984).

This structure enables one to create a feature that is, for example, a point feature (when examining spatial attribute 1, say), a line feature (spatial attribute 2), an area feature (spatial attribute 3), or a set of features (spatial attribute 4). An example might be an airport digitized as a point feature at a small scale, as a line feature showing the main runway at a larger scale, or as an area feature at an even larger scale. Finally, the feature may be defined by its constituent runways, hangars, control towers, etc., at a very large scale.

Although this use of a vector of spatial attributes is not the only use (another example may be different views of a feature as digitized from photographs taken in different wavelengths), it is the motivating use for having a vector of spatial attributes. This vector is thus often referred to as detail levels.

Associating each detail level with a scale range provides a mechanism for displaying only an appropriate amount of detail at a certain scale, that is, generalization.

Efficient spatial attribute manipulation is obtained by imposing a quad-tree index on the spatial attributes. The set of quad-tree leaves containing each node, chain and region is computed according to a scheme described by Abel and Smith (1982). Using only the leaves of the quad-tree imposes some storage overheads, but decreases the response time to spatial queries.

Non-spatial attributes. Four requirements must be satisfied by the non-spatial attributes of a feature. These attributes must provide:

1. a classification scheme for feature coding;
2. the type of a feature at each detail level;
3. the allowable set members of a compound feature;
4. definitions of other descriptive attributes.

These requirements may be met by constructing a poset of classes and subclasses, each feature classification being a complete path name in the poset. Figure 2 gives an example of a poset defining some classes for cultural features. The vertices are distinct feature classes, and the arcs denote a major class/subclass relation. A path such as "cultural features/aviation/airport" may be codified as an integer and used as a feature classification. Some vertices in the network may not be complete classifications (for example, "cultural features/aviation"), and hence a distinction is made between these two types of vertices. If a vertex completes a classification then it has associated with it the definition of the feature types at each detail level of a feature of this class, as well as an indication of which compound features the feature may be a part.

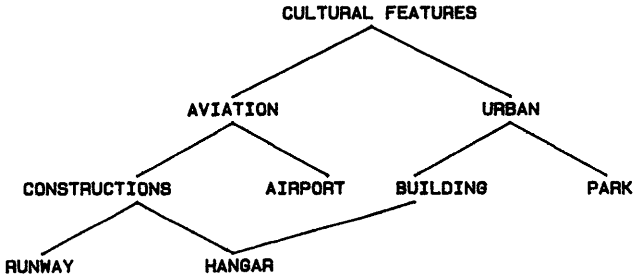


Figure 2: Example subset of the classification of 'Cultural Features'

Finally, each vertex in the poset may have a number of other associated attributes. Subclasses (that is, classes lower down in the poset) inherit all attributes of their major classes. All classes have the same virtual class as a major class. This class associates all attributes common to all features, for example feature name. The attributes may have **computed** or **constraint** clauses defined on them, stating the interactions of the attributes. An example of how some of the classes in Figure 2 may be coded follows. Comments are written following a double hyphen.

```
Cultural Features refines Class  -- No further information of this
end.                             -- subclass is required.  It has no
                                 -- other spatial or non-spatial
                                 -- attributes.

Aviation refines Cultural Features
end
```

```
Airport refines Aviation;          -- This completes a feature
  Spatial: point at 1;             -- classification and thus includes
           line at 2;              -- information on the spatial
           area at 3;              -- attributes for each detail level.
           set at 4;               -- The computed clause specifies how
  Non-spatial: area : numeric;     -- area may be calculated from the
    computed at 4 :                  -- area attributes of elements of
                                   the
      sum (airport.4.set = area);-- composite feature.
end.

Hangar refines Constructions, Buildings; part of Airport;
  Spatial: point at 1,2,3;          -- The part of clause specifies
           area at 4;               -- that a hangar may be included
  Non-spatial: area : numeric;      -- in the set of airport
end.                                -- composite spatial attributes.
                                    -- The area attribute is
                                    -- referred to by the computed
                                    -- clause in the airport class.
```

Implementation of this scheme borrows concepts from object-oriented languages.

TOPOLOGY

The topology of a GIS contains information about the structure of spatial attributes and is independent of the form of the attribute. Three types of topological relationships are discussed: incidence, containment and exclusion.

Incidence

Incidence information between nodes and chains (0-1 incidence) and between chains and regions (1-2 incidence) is kept. Node-region (0-2) and chain-chain (1-1) incidence may be derived from these owing to the transitive nature of the incidence relation. Some of this information is already available from the construction of the database (specifically chain→node and region→chain relations). The other information (node→chain and chain→region) is coded explicitly. The use of incidence relations has been discussed extensively by White (1984).

Containment

Containment information is required to answer geographical queries. The relations required are nodes inside regions (0-2 containment), chains inside regions (1-2 containment) and regions inside regions (2-2 containment). Although one of the design objectives is to keep all topological information separate from the spatial attributes, this can be relaxed for containment if one assumes that no projection (or map deformation) will map a node or chain either into or out of a region. This assumption is not unrealistic since most projections are continuous functions under the normal distance metric in n-space. Under this assumption containment can always be calculated, at the obvious expence of response time.

Exclusion

For metric queries concerning area features, and in order to render regions correctly during cartographic representation, it is necessary to keep information about regions excluded from other regions. Since nodes and chains have no area, they need not be considered for the exclusion relation. The maintenance of this relation results in an extension to the definition of area features. Each area feature, in addition to a number of included regions, also has a number of excluded regions for each included region.

In contrast to containment, exclusion cannot be calculated since it is a semantic concept, not a spatial one.

Clearly, the topology on a set of spatial attributes is dependent on the actual types of the spatial attributes of features (that is, point, line or area). This means that it is only sensible to make topological queries on a single detail level, since no topology is defined across detail levels. This constraint may be enforced by the user interface of the GIS.

CONCLUSION

A number of objectives for a geographic database and a design that meets these objectives have been described. A prototype of the database using a simplified strictly tree structured hierarchical classification scheme has been implemented using a commercial database management system.

Further work will include the implementation of the complete poset scheme for the management of non-spatial attributes, and the extension of the concepts to include multi-user access without compromising the integrity of features or topology.

REFERENCES

Abel, D J and Smith, J L, 1982. "A Data Structure and Algorithm Based on a linear Key for a Rectangle Retrieval Problem", Computer Vision, Graphics, and Image Processing 24, 1 - 13.

Guptil, S C, 1986. "A New Design for the US Geological Survey's National Digital Cartographic Data Base", Procedings of Auto Carto London, Vol 2, 10 - 18.

Peuquet, D J, 1984. "A Conceptual Framework and Comparison of Spatial Data Models", Cartographica, Vol 21 No 4, 66 - 113.

White, M S, 1984. "Technical Requirements and Standards for a Multipurpose Geographic Data System." The Americal Cartographer, Vol 11, No 1, 15 - 26.

THE dbmap SYSTEM

Donald F. Cooke
Geographic Data Technology, Inc
13 Dartmouth College Highway
Lyme, New Hampshire 03768

Beman G. Dawes
RD #2, Box 35
Onancock, Virginia 23417

ABSTRACT

The dbmap system consists of a high-level language for specifying digital mapping and geographic information systems and a compiler/execution monitor to run an application programmed in dbmap. The system is readily expandable due to an open-architecture design, and produces applications with very modest execution overhead. dbmap currently runs on IBM PC-AT computers and can probably be ported to any system with an ANSI "C" compiler. Operational systems written in dbmap have proven their worth in over six months of testing in production environments.

BACKGROUND

After six years of writing monolithic computer programs to support map digitizing, updating and editing operations, we designed and implemented a language -- "dbmap" -- expressly intended for generating a wide variety of application packages. We had become frustrated with maintenance and modification of an increasing number of operational systems, each of which needed support from an experienced programmer. We were also faced with opportunities to market diverse map-related software systems applied to vehicle routing and dispatching, map updating, geocoding and spatial analysis.

We had rejected conventional approaches to "generalized" mapping such as systems supplied by Intergraph or E.S.R.I. for several reasons: first of all, in 1980, none were addressing topological data structures explicitly. Secondly, such systems sacrificed execution efficiency for generality and flexibility. Finally, they could not be used as the foundation for inexpensive added value products because of their high cost and the price of the hardware they required.

In January 1986, we started a design process which ran for three months. We rejected both previous monolithic programs and the ARC-INFO/Intergraph paradigm of a flexible, tailorable system. Instead we settled on a simple, powerful language capable of specifying the characteristics of a desired system. The language, called "dbmap", is simple to learn and extremely flexible.

A junior programmer can learn dbmap in a few days, especially if there are a variety of examples to follow. As with other languages, the dbmap language statements must be keyed into a command file which is read by the dbmap compiler.

The dbmap compiler generates "P-code" which is immediately executed.

The dbmap compiler is currently written in ANSI "C". We expect to run dbmap primarily in IBM PC-AT and '386 environments. Our experience with running the "db" subset (database management only, no graphics) on a Data General MV-4000 suggests that dbmap would perform well on Macintosh, VAX or other machines.

In general, we designed for the fast single-user systems we expect will prevail in the near future. A typical minimum configuration that we run in daily operation is a 640K PC-AT clone running at 8 Megahertz. We configure such a machine with a fast 40 Mbyte Winchester disk, a mouse and two monitors: EGA for map display and Hercules monochrome for text. Cost per workstation at "street price" is about $3500.

APPROACH

We developed dbmap using traditional software engineering methodology:

- Problem analysis
- Requirements definition
- Overall system design
- Component design, implementation, and testing
- System level testing
- Refinement and maintenance

As in other non-trivial systems, the actual process was often more interative than sequential. Two steps forward were often followed by one or two steps back.

INTEGRATED SYSTEM

dbmap is a single integrated software system rather than a series of separate application systems because of interaction between applications. For example, address matching (geocoding) is improved by the capability of displaying a map of matched addresses.

Consequently, we implemented dbmap as a single large program able to perform almost one hundred separate functions. Some of the functions are simple, such as changing the color of the cursor. Others are of medium complexity, such as performing an index file search. Very complex functions include creation of a complete database through a screen manager.

PROGRAMMING LANGUAGE

dbmap is designed to be programmable rather than fixed because the details of how an activity is carried out vary according to the context. For example, digitizing nodes or shape points in a Census Bureau TIGER database is a very different context from digitizing address matching rejects though the underlying digitizer functionality is the same.

dbmap therefore is a high-level programming language with facilities for data definition, execution sequence control and expression evaluation. These facilites can be used both procedurally, like Pascal, FORTRAN or "C" programs, or declaratively to specify database and other parameters.

Execution of a dbmap application requires a dbmap language compiler/execution monitor which is described below.

OPEN ARCHITECTURE

We designed dbmap as an open system since we did not feel we could anticipate all required functions at design time. In the event that a new function (converting Arabic numbers to Roman numerals, or decimals to fractions) is needed, it is possible to define and write the function in such a way that the dbmap system makes it available to all dbmap applications.

dbmap's open software architecture presently requires that new functions be written in "C". A dbmap function begins with a control block that specifies dbmap language formal argument requirements to the dbmap compiler. This scheme allows the dbmap compiler to handle an ever-expanding library of functions as if all functions had been built into the language when the system was designed.

High level functions such as input/output, database indexing, graphic and geographic operations are not built into the dbmap language but have been implemented via the open architecture methodology.

New functionality can be added either through "value-producing" functions (similar to subroutines) or at a much higher (superstructure) level which allows creation of a complex environment like a sort/merge utility or report generator.

DATABASE MODELS

dbmap supports a variety of database structures because no single database design can serve all applications. For example, (1) the network database model (2D) works well for DIME or TIGER file creation and maintenance, (2) an unnormalized relational model (the traditional 300-byte DIME file) is better as a distribution format and (3) a point database is adequate for supporting centroid-based retrieval systems like "On-Site" and "Area Profile Reports". Route optimizing or choropleth mapping may require still other structures.

In internal memory, dbmap often uses a network model, but external disk files may be any structure. In general, the needs of the application determine the file structure.

OPERATING ENVIRONMENT

We required flexibility in hardware and operating system environments for dbmap applications since no single environment can serve all potential uses. For example, spatial spreadsheet (desktop GIS) manipulation calls for a personal computer, map digitizing is a workstation application and large-scale address matching is still a mainframe operation. Applications on the new Apple, Amiga and Atari machines (not to mention the forthcoming CD/I systems) look increasingly attractive.

The only common thread in this mix of environments is the likelihood of availability of a "C" compiler. dbmap was implemented using the draft proposed ANSI "C" language. Considerable care was taken to insure portable code, with initial development done on AT-Clone microcomputers. Subsequently the non-graphics portions of dbmap have been moved to a Data General MV-4000 minicomputer; all code compiled and ran correctly on the first attempt and has been used in that environment for several months.

TRADEOFFS

Operating speed is of considerable importance to dbmap users for economic reasons and ease-of-use. Many users will spend much of their workday using dbmap, so slow response or error prone applications are not acceptable.

Ease of programming is not as important as flexibility and ease of use. For every person hour spent programming applications in dbmap, hundreds of person hours will be spent using the applications. On the other hand, programmers must be much more productive in dbmap than in languages like C or Pascal to avoid traditional software bottlenecks - and these same dbmap programmers will likely be less experienced than typical C or Pascal programmers!

The dbmap compiler uses a compile and go approach similar to commercial "turbo" compilers. Thus a dbmap program is compiled into memory each time it is used. Compile time is less than half a second for simple dbmap programs, and less than ten seconds for very large programs.

The fast combined compile/test cycle aids dbmap programmer productivity.

Compiling each use implies that the latest version of dbmap functions are always invoked. Thus if a bug is fixed or an algorithm is improved, the benefits apply right away to all applications which use the function.
Compiling rather than interpreting results in acceptable operating speed. For example, complex data processing jobs commonly run at 100,000 to 200,000 records an hour on an AT-clone while simple sequential file searches run at over 1,000,000 records per hour. File operations actually run acceptably fast on a PC-clone but graphics operations need the speed of an AT at a minimum.

"ZONE RANGER" -- A dbmap APPLICATION

The first system written in dbmap was one we call "Zone Ranger". We use Zone Ranger to create custom address coding guides for computerized dispatching services. Customers indicate their delivery territories (zones) on standard maps, usually USGS 7.5 minute quads, by outlining the zone boundaries with a marker. We display corresponding images of our digital maps on Zone Ranger's EGA monitor and use the mouse to "lasso" each zone in turn. When all zones have been lassoed (and line-segments in our data base tagged with zone numbers) a utility program creates the customer's address coding guide in deliverable format.

Figure 1 illustrates the Zone Ranger CRT display in a monochrome implementation used for these figures. The main window shows a zoomed image of the New York City financial district selected from the Manhattan's 8753 line segments previously loaded into dbmap's working RAM. The operator has used the "Name" function to label ten streets.

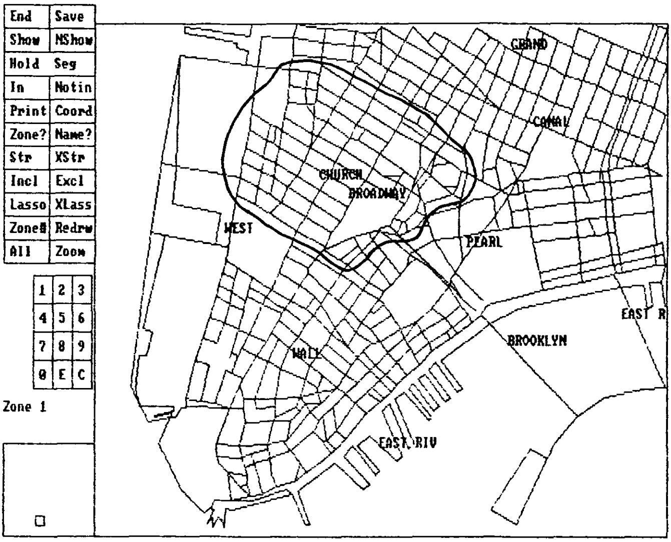


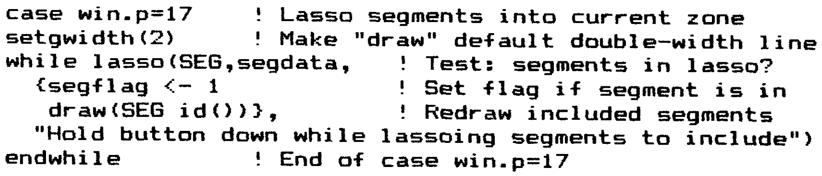
Figure 1 Zone Ranger CRT display

On the left of the screen from top to bottom are the 22-item main function menu, a "soft" keypad for entry of zone numbers, the current zone identifier and a small orientation map that confirms that we've zoomed in on a small area in the south of the submodel.

The operator can toggle line segments in or out of the current zone one at a time (Incl & Excl), a street at a time (Str & XStr) or by using the mouse to lasso a group of segments (Lasso & XLass). Figure 1 shows the lasso trace of a delivery zone northeast of the World Trade Center.

In this example the operator activated the lasso function by clicking the mouse in the "Lasso" menu box, the 17th pane of the main function window. This action activates case "win.p=17" of the dbmap language code for the Zone Ranger application (Figure 2).

Figure 2 shows all of the dbmap code needed to specify the lasso capability. Line 1 identifies the code for clicking on the 17th pane of the main menu; an exclamation point delimits a comment field. "setwidth(2)" invokes a dbmap function that makes subsequent calls to "draw()" plot

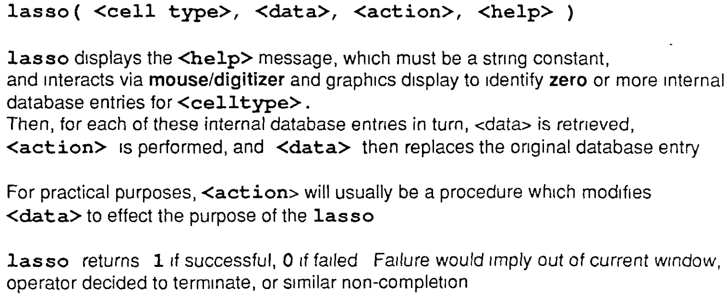


```
case win.p=17      ! Lasso segments into current zone
setgwidth(2)       ! Make "draw" default double-width line
while lasso(SEG,segdata,    ! Test: segments in lasso?
  {segflag <- 1             ! Set flag if segment is in
   draw(SEG id())},         ! Redraw included segments
  "Hold button down while lassoing segments to include")
endwhile           ! End of case win.p=17
```

Figure 2 dbmap Code for Zone Ranger Lasso Function

double-width lines. (In the EGA color version of Zone Ranger we choose a line color instead of width.)

Lasso needs four parameters:



lasso(<cell type>, <data>, <action>, <help>)

lasso displays the **<help>** message, which must be a string constant, and interacts via **mouse/digitizer** and graphics display to identify **zero** or more internal database entries for **<celltype>**.
Then, for each of these internal database entries in turn, <data> is retrieved, **<action>** is performed, and **<data>** then replaces the original database entry

For practical purposes, **<action>** will usually be a procedure which modifies **<data>** to effect the purpose of the **lasso**

lasso returns **1** if successful, **0** if failed Failure would imply out of current window, operator decided to terminate, or similar non-completion

Figure 3 "Lasso" Function Definition from dbmap Manual

The first parameter says we're interested in lassoing line segments in the database, not nodes, points or other objects. "segdata" is the name of a data control block for line segment data, one element of which is "segflag" which if set indicates inclusion in the current zone.

"<action>" happens to each segment determined to be inside the lasso: "segflag" gets set to 1 and the segment is redrawn, now with double line width. The help message appears on top of the main map screen as the lasso is being drawn. (The help message doesn't show in Figure 1 which really shows the "Print" function.)

Figure 4 shows the Zone Range screen after the lasso operation.

CONCLUSION

It's probably clear from the example that dbmap is more difficult to program than a general database system like dBase III or RBase 5000. We feel that dbmap's ability to handle the more complex world of computer cartography compensates for and explains its demands on the programmer. We have found that junior programmers can become conversant with dbmap in a week or two and can master implementation of a new system after a month's experience.

A year ago we had not begun implementation of dbmap; at present we have several production systems in operation in-

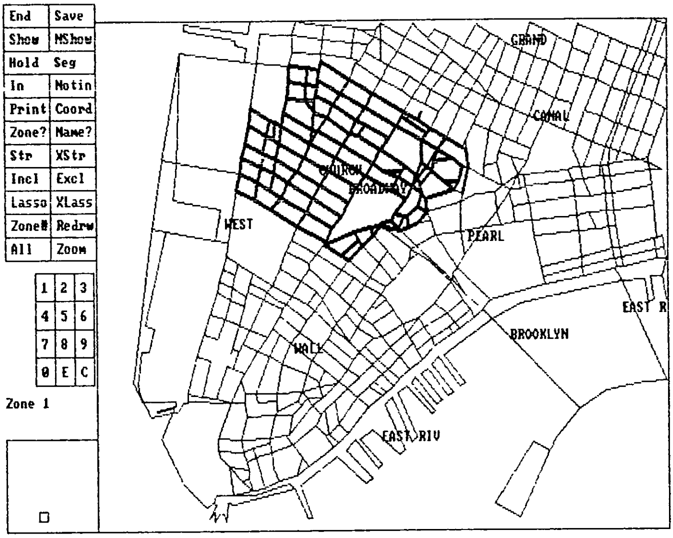


Figure 4 Result of Lasso Operation on Figure 1

house and at customers' sites. We are unequivocally pleased with both the dbmap language and the applications developed so far. We expect dbmap applications to supply all our internal mapping needs by mid-summer, 1987 and to provide the foundation of a family of integrated mapping products.

ACKNOWLEDGMENTS

The authors wish to thank Robert Reeder, Ted Jerome and Patrick Ferraris for their valuable suggestions and contributions.

DESIGN AND IMPLEMENTATION OF MICROCOMPUTER BASED WATER RESOURCES DECISION SUPPORT SYSTEMS

Marc P. Armstrong
Departments of Geography and Computer Science
316 Jessup Hall
The University of Iowa
Iowa City, IA 52242
BLAMMGPD@UIAMVS.BITNET

ABSTRACT

A general design strategy for building water resources decision support systems is outlined. The framework uses a modular representation of system functions which must be blended for successful system implementation. Specific emphasis is placed on data organization strategies.

INTRODUCTION

Water is an important resource. Because of its importance, water management practices are commonplace in many arid regions of the world. Increased attention is now being placed on managing water resources in humid regions, as well. This attention has had its genesis in two factors: 1) a growing recognition that water supplies are finite even in areas with seemingly vast stores of fresh water (Cohen, 1986); and 2) an acknowledgement of the role that pollution plays in the long-term viability of both ground and surficial water holdings.

As a result of these concerns, need has increased for timely information upon which to base water management decisions. Researchers have responded by implementing information systems built specifically to address water management issues. Some systems have been designed to support decisions about surface water (Guariso, et al., 1985; Johnson, 1986; Holsapple and Whinston, 1976; Hopkins and Armstrong, 1985), while others deal with ground water resources (Hendrix and Buckley, 1986; Monaghan and Larson, 1985). Although these individual systems are meritorious, the implementations are unrelated, and seem to employ an ad hoc approach to system design. The purpose of this paper is to resolve this problem by describing an overarching strategy for designing and implementing a microcomputer based water resources decision support system (DSS). By exploiting inexpensive microcomputer technology, the strategy may prove palpable to system designers and decision-makers charged with solving water management problems in both developed and developing countries.

The paper is divided into two main sections. The first section is concerned with developing a general framework for designing a water resources DSS. The second provides a consideration of specific issues pertaining to an example application of the framework.

A DESIGN FRAMEWORK FOR WATER RESOURCES DSS

The goal of a DSS is to help decision-makers in the processes of solving structured and, more importantly, semi-structured problems. Many spatial problems are semi-structured or ill-defined (Hopkins, 1984) because all of their aspects cannot be measured or modeled. This aspect of semi-structured problems necessitates human intervention, and therefore, solutions to semi-structured problems are often obtained by allowing a decision-maker to select and evaluate workable solutions from a set of alternatives. These steps are followed in an exploratory and sometimes heuristic fashion until an outcome acceptable to the decision-maker is reached. The system, therefore, must employ feedback loops to allow the user to evaluate the usefulness of solutions, and perhaps, to alter model parameters, or even to choose entirely different modeling strategies. To achieve these objectives, decision support systems normally use a variety of data types, and also rely on graphic displays to convey information to the decision-maker. Many systems also incorporate artificial intelligence principles to make them easy to use.

A DSS can be constructed from a set of linked software modules (Armstrong, Densham and Rushton, 1986). In a water resources DSS, the modules and the data stored within, can be organized in many ways depending upon institutional objectives and the nature of decisions that a system is designed to support. Despite this potential for organizational diversity, common design principles can be adhered to during the construction of any system. The framework described here has been adapted from Sprague and Carlson (1982). Its main components are:

1) Geometric Representations
2) Operations
3) Structure
4) Mechanism for Interaction.

Although each can be considered to be equally important, in this paper, the first three components will be discussed, and particular emphasis will be placed on the third (structure) component. It must be stressed that the ultimate objective of the system designer is to create a seamless, rather than modular, view to the DSS user. These modules, therefore, need not be separated in a real sense. Viewing the system in this way, however, facilitates software development tasks.

Geometric Representations

At the DSS design stage, choices must be made about topological referencing methods, and the degree of spatial precision used for analytical operations and for storing cartographic representations. A two-tiered approach to the organization of these data, such as that described by Hopkins and Armstrong (1985), can provide a flexible means for accommodating the topological and cartographic data. Hopkins and Armstrong, however, were concerned with a stream channel information system. The structure presented here is more general, in the sense that it explicitly accommodates interfluve information that is often critical in water management decision-making. Other relationships such as flow distance, are also readily accommodated in the two-tiered approach.

In this tiered approach, the main design elements are the stream channels, rather than the basins, because of an underlying need to efficiently specify and retrieve flow relationships. This main organizational tier forms a topological skeleton and provides for macro referencing capabilities with respect to the hydrological network. Although the skeleton provides a useful structural mechanism for general representations of database entities, the second, cartographic, tier provides their explicit descriptions; each·topologically referenced entity has coordinate information that is requisite for display and analytical functions.

Operations

The number and types of operations in a water resources DSS are controlled by a need for information upon which to base decisions, a need to select from alternative problem solving strategies, and a need to provide effective representations. Among the analytical operations often needed in a water resources context are:

- Production of summary statistics. These data are used in the course of producing environmental inventories and assessments for basin planning.

- Application of logical decision rules. Operations of this type are used to determine suitabilities from combinations of variables. The results are often used in assessing the impact of proposed development projects, and for basin planning.

- Hydrological modeling capabilities are important components of a water resources DSS, because they provide a mechanism for performing exploratory analyses. For example, by changing runoff parameters, impacts on hydrological characteristics can be determined for various development scenarios.

- From a cartographic standpoint, important operations allow simplification (Douglas and Peucker, 1973) or enhancement (Dutton, 1981) of stream traces for producing thematic maps at various scales.

These functions, and others, are obtained by retrieving and manipulating geometric and thematic information contained in the database. These data are then passed to modules designed to produce cartographic displays, graphs, and formatted reports. Operations are vital to a water resources DSS, because they provide the user with a tangible basis for validating decision-making outcomes.

Structure

The way in which information is organized in any computer system is a critical factor in its success or failure. The chosen structure must provide a means for capturing the fidelity of data relationships that must be accessible to solve either individual problems, or entire classes of problems. The storage structure also influences the user's conceptualization of the database, which in turn, influences the types of problems that a user will attempt to solve. At a most fundamental level, the implementation of the user view plays an important role in system performance.

The structure component of the water resources DSS design framework takes the form of a detailed database design and implementation strategy. An important component of a database is the adoption of a logical model to support representations and operations. Logical models vary in the types of data relationships that they support, and differ in methods for producing efficient linkages among database elements. The major logical database models can be placed into two families: operations-oriented (e.g. relational) and structure-oriented (e.g. network).

Miller (1984) has provided a structure based upon the operations-oriented relational model (Codd, 1982). The relational model, as it is now often implemented for microcomputers, may be unsuitable for DSS application development. Retrieval performance is slow, compared to alternatives, and it requires storage of redundant normalized data domains. Many microcomputer implementations of the relational model also are limited in their joining capabilities when compared to mainframe versions. Other problems with a purely relational approach to data modeling are recounted elsewhere (King, 1981; Sandberg, 1981).

Hopkins and Armstrong (1985) provide a water resources database structure that uses a network design. The network model employs fixed linkages to provide a mechanism for forming relationships among database entities (Olle, 1978). Paths specified by the database designer are used during retrievals. Although the network design is efficient with respect to the relational approach for retrieval types that are known to the database designer, performance may be degraded when alternatives unanticipated by the designer must be explored. The path dependencies then become a liability rather than an asset. Better alternatives are available.

In this paper, I provide a structure based upon the extended network model (Bonczek, Holsapple, and Whinston, 1976; 1984), a hybrid model that exhibits the retrieval performance characteristics of the network model, while providing much of the flexibility of the relational model. The extended network model bears some similarities to the network model; it differs mainly in implementation. Both models use set relationships among record entities in the database. The extended network model, however, provides for a number of advanced logical structuring capabilities that are especially useful for spatial data processing applications: many-to-many sets, recursive sets, and system-owned sets.

Many-to-many sets. The extended network model allows the direct specification of many-to-many (N:M) linkages between database elements. The direct, and thus, efficient, provision of N:M sets is useful, because spatial databases often contain entities and attributes that are linked inherently in many-to-many relationships. For example, coordinate chains constitute the piecewise approximation of more than one polygon (e.g. a shared border), and can be owned directly by both polygons in the extended network structure. The database designer or user, therefore, need not be concerned about the specification of polygon-chain-node pointer structures.

Recursive sets. Extended network structures also provide for recursive relationships, wherein records of a given type can own other records of the same type without having to traverse additional paths in an ownership tree structure. This feature is useful when describing topology, because it obviates the need for separate contiguity, or flow, record structures. For example, a water resources DSS must be able to support a series of data structuration capabilities that will permit rapid retrieval of flow relationships including: upstream, downstream, or tributary determination. Although these relationships could be calculated from three dimensional coordinates, that process is time consuming and error-prone. Because they are often invariant over the life of a database, hydrological relationships are easily determined from maps and stored in a recursive set. A set relationship of this type is formed when data are added to the database. The linkages are not computed "on the fly" (e.g. joins) as they are in operations-oriented approaches to the same problem.

System sets. The extended network model allows independent direct access of any record type by simply declaring it to be system-owned. This obviates the need for chaining through intermediate record types to retrieve information about database entities. If a simple spatial hierarchy, such as streams and sampling stations, is created, stations can be made members of a system-owned set. It is not necessary, therefore, to know the stream on which a station is located to retrieve information about that station. Note, however, that the original hierarchy can also be retained and used if, for example, it is necessary to determine all stations on a single stream.

Ease of retrieval can be gained by declaring many system-owned record types, and comes with only a minor penalty of incurring increased overhead storage (about four bytes per link) for each instance of a system-owned set in the database (Bonczek, Holsapple, and Whinston, 1984:107). This facility helps to provide a tabular, or relational-like view of the database.

DSS DESIGN APPLICATION

In this section an example application is outlined. It draws upon the design strategy from the previous section, and employs capabilities of the extended network model.

First, a general schema diagram is used to illustrate logical relationships in the database. Then a portion of an example schema is specified in a data definition language (DDL).

Schema Diagram

In Figure 1, the main organizational entity is the stream. Each stream, however, can be accessed in many ways to increase flexibility in terms of both jurisdictional referencing (e.g. stream identification for different governmental agencies) and the human interface (e.g. stream name). Note that two recursive sets are present for each stream - one each for tributaries, and when required, distributaries. These sets provide an effective means for encoding flow relationships.

A stream also has precedence over other entities (nodes, lines, areas) that exist either wholly or partially within the areal extent of its basin. Examples of these entities are: wells, transmission lines, and recreation areas. When data are organized in this way, entities are explicitly assigned to basins. Entities that extend across basins, however, can be handled by many-to-many relationships.

Each entity (e.g. a well) also explicitly owns its geometrical description in the form of chains or points. The use of many-to-many sets is a convenient way to structure chain-encoded polygon data. Each polygon owns many chains; each chain is owned by many (two) polygons. Likewise, each chain has two nodes (from, to) and each node, by definition, serves as a terminator for many chains.

In this general structure, each entity can have many attributes. For example, stream entities may have several bridges associated with it; it may also have information about a multitude of gauging stations, historical sites, and recreation areas. In Figure 1, these attributes are sorted by distance along the stream (RMI), by bank (left, right, both, instream) and by date. Of course other strategies exist; these are meant to be illustrative.

Data Definition

After the relationships among database elements have been designed in graphic form, they must be coded in a DDL (Figure 2) prior to implementation. In this example, the DDL syntax of MDBS III (Bonczek, Holsapple and Whinston, 1984) is used. It provides a rich database environment for a variety of microcomputer systems, and supports the extended network model. The intent here, is to provide the flavor of how a DDL specification is constructed; space limitations preclude a total description.

SUMMARY

A design framework for decision support systems can be adapted readily to water resources applications. The logical structuring facilities of the extended network model support the geometrical and operations requirements of the water resources DSS, and provide for data organization in a single, unified repository. Stream flow (topological) relationships are specified using recursive sets. Cartographic representations are stored using many-to-many sets. Attribute information is organized by date, bank, or along the linear dimension of a stream.

REFERENCES

Armstrong, M.P., Densham, P.J., and Rushton, G. 1986. Architecture for a microcomputer based spatial decision support system. Proceedings, Second International Symposium on Spatial Data Handling. Williamsville, NY: IGU Commission on Geographical Data Sensing and Processing.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. 1976. Extensions and corrections for the CODASYL approach to data base management. Information Systems, 2: 71-77.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. 1984. Micro Database Management: Practical Techniques for Application Development. Orlando, FL: Academic Press.

Codd, E.F. 1982. Relational database: A practical foundation for productivity. Communications of the ACM, 25: 109-117.

Cohen, S.J. 1986. Climatic change, population growth, and their effects on Great Lakes water supplies. Professional Geographer, 38: 317-323.

Douglas, D.H. and Peucker, T.K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, 10: 112-122.

Dutton, G.H. 1981. Fractal enhancement of cartographic line detail. American Cartographer, 8: 23-40.

Guariso, G., Rinaldi, S., and Soncini-Sessa, R. 1985. Decision support systems for water management: The Lake Como case study. European Journal of Operational Research, 21: 295-306.

Hendrix, W.G., and Buckley, D.J.A. 1986. Geographic information system technology as a tool for ground water management. Technical Papers, ACSM-ASPRS Annual Convention. Falls Church: American Congress on Surveying and Mapping.

Holsapple, C.W. and Whinston, A.B. 1976. A decision support system for area-wide water quality planning. Socio-Economic Planning Sciences, 10: 265-273.

Hopkins, L.D. 1984. Evaluation of methods for exploring ill-defined problems. Environment and Planning B, 11: 339-348.

Hopkins, L.D., and Armstrong, M.P. 1985. Analytic and cartographic data storage: a two-tiered approach to spatial decision support systems. Proceedings, Seventh International Symposium on Computer-Assisted Cartography. Washington, DC: American Congress on Surveying and Mapping.

Johnson, L.F. 1986. Water resource management decision support systems. Journal of Water Resources Planning and Management, 112: 308-325.

King, J.M. 1981. Evaluating Data Base Management Systems. New York: Van Nostrand Reinhold Co.

Miller, S.W. 1984. A spatial data structure for hydrologic applications. Proceedings, International Symposium on Spatial Data Handling. Zurich: Geographisches Institut, Universitat Zurich-Irchel.

Monaghan, G.W. and Larson, G.J. 1985. A computerized ground water resources information system. Ground Water, 23: 233-239.

Olle, T.W. 1978. The CODASYL Approach to Data Base Management. New York: John Wiley and Sons.

Sandberg, G. 1981. A primer on relational data base concepts. IBM Systems Journal, 20: 23-40.

Sprague, R.H. and Carlson, E.D. 1982. Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice-Hall Inc.

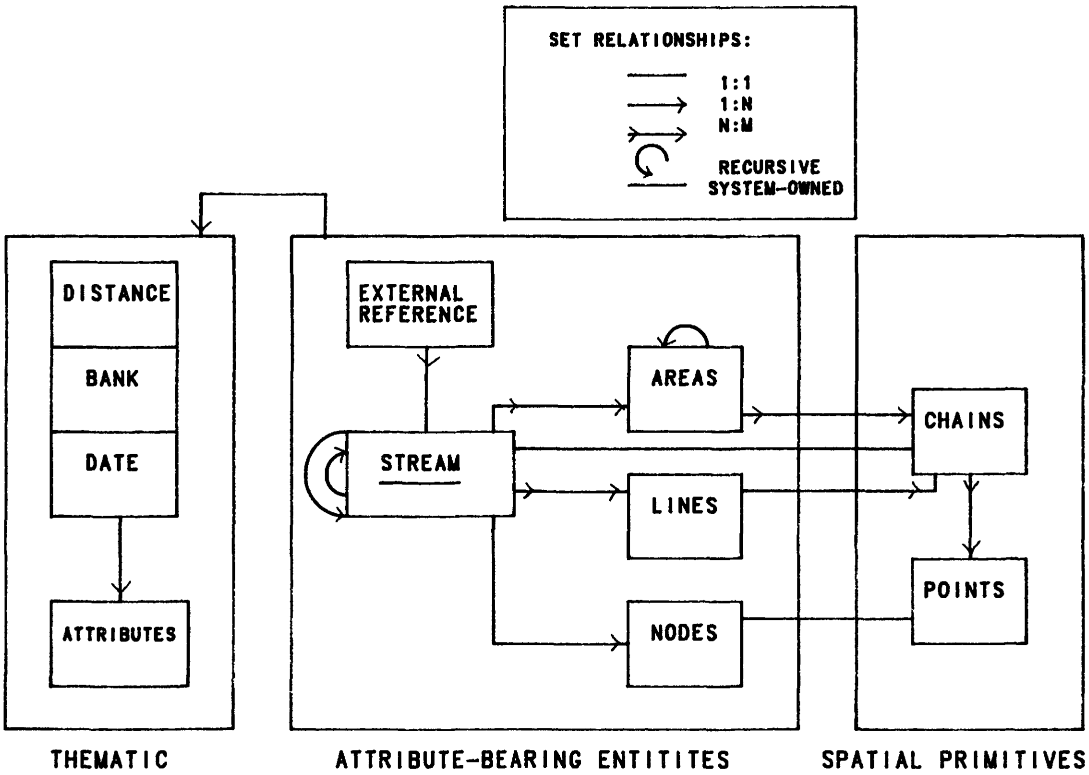


FIGURE 1: DATA ORGANIZATION FOR WATER RESOURCES DSS.

```
/*******  DATABASE  IDENTIFICATION AND SECURITY   *******/
/*                                                      */
database name is STREAMS
user is Granite with ROCK
user is Pillsbury with ROLL
/*                                                      */
/*******           RECORD SPECIFICATION        ********/
/*                                                    */
record name is STREAM
       item name is STREAMNAME string 20
       item name is DRAINAREA real 2
       item name is INFLOWRMI real 2
       item name is STREAMLEN real 2
record name is PSEUDONYM
       item name is STNAME  string 20
       item name is OTHERID  string 20
record name is WELL
       item name is STPLANEX real 3
       item name is STPLANEY real 3
       item name is FIPSCO unsigned 1
       item name is SECTION unsigned 1
       item name is TOWNSHIP string 3
       item name is RANGE  string 3
       item name is TOPELEV real 2
       item name is DEPTH real 2
       item name is H2OLVL real 2
/*                                                       */
/********           SET SPECIFICATION           ********/
/*                                                      */
set name is STREAMS  type is 1:N
       owner is SYSTEM
       member is STREAM order is FIFO
set name is TRIBS type is 1:N
       owner is STREAM
       member is STREAM order is FIFO
set name is EXREFS type is N:1
       owner is PSEUDONYM sorted ascending by OTHERID
       member is STREAM
set name is DEEPSUBJECT type is 1:N
       owner is STREAM
       member is WELL sorted ascending by COFIPS
set name is WELLDIR type is 1:N
       owner is SYSTEM
       member is WELL sorted ascending (TOWNSHIP, RANGE)
end
```

Figure 2. Schema definition in DDL.

TRENDS IN HARDWARE FOR GEOGRAPHIC INFORMATION SYSTEMS

Jack Dangermond
Scott Morehouse
Environmental Systems Research Institute
380 New York Street
Redlands, CA 92373

ABSTRACT

This paper presents a description of and comments on various trends in the hardware available for Geographic Information Systems (GIS's). After a brief introduction the paper deals with fast geoprocessing hardware, parallel processing, memory, workstations, networks, and hardware for specialized processing functions. Finally there are some comments on the UNIX operating system and on various peripheral devices including the need for new kinds of specialized workstations for GIS use.

INTRODUCTION

This paper describes and discusses some of the trends which we see in the hardware used in Geographic Information Systems (GIS's). We think such a paper may be useful because of the major impact that hardware developments have on geographic information system architecture, on GIS software, on costs and efficiency and on the whole nature of the user interface with GIS's. Hardware is certainly one of the three major drivers of developments in GIS's and probably the one in which change continues to occur most rapidly.

FAST GEOPROCESSING

There is now some interest in such tasks as processing geographic data for all of the North American continent or weather information for the whole world in near real time. Over the next 10 years CRAY-type machines with the speed and performance necessary for such tasks are going to become available at affordable prices, and1,000 MIPS CPU's within large organizations will become common.

Providing one or two network stations at people's desks, just to satisfy their need for interactive display and editing, doesn't solve the problem of data analysis for very, very large geographic data sets. In five years 20 to 30 MIPS workstation processors are likely. There is already discussion of 100 MIPS PC's in the early 1990's for about the same price paid for a PC today. At that time100 to 1,000 MIPS processors that are the file servers are also likely, and in the price range of today's minicomputers .

The implication for software developers is that they need to begin right now to develop the algorithms for that hardware environment; they need to concentrate on doing it correctly and not worry so much on whether it takes 5 minutes or 1 minute to actually perform a particular operation. They need to consider that performance with large databases is the real issue, not speed. Of course algorithms should be written economically and efficiently, but the spending of hundreds of hours to save minutes of time on a particular process is going to be of less interest given the state of the art of the hardware and the direction that hardware development is going.

PARALLEL PROCESSING

Even with 1,000 MIPS machines there will still be limitations on the performance of individual computing units based on the amount of circuitry that can be packed into a small space, the speed of light and the need to dissipate heat from the circuitry.

Because of these limitations, further dramatic breakthroughs in the performance of serial processing computers are not certain and order of magnitude increases in speed and computing power may have to be based on the use of parallel processing in which the computational task is divided among many processors. Even given a single task like generating a map or computing a polygon overlay, algorithms which inherently support parallel processing will benefit. Different computer manufacturers are now developing prototype multiple parallel processors and are beginning to provide support for parallel processing operating systems.

This development means that the software will have to change to take advantage of the parallel processing. There are two ways to handle true parallel geoprocessing. One is that the operating system breaks the problem into pieces, does all the optimization, does the sorting in one box, does each of the other needed operations in other separate boxes. The other approach is to program specialized parallel processors that take advantage of the fact that geography can inherently be spatially subdivided: for example, each of a hundred processors takes one-hundredth of the problem and does the polygon overlay on just that piece. This approach will require that we reorganize the way GIS software runs. The tiling structure in ESRI's map library system might be a suitable beginning point for such an approach: each of the map tiles would be dealt with by a single processor. Geoprocessing can also be viewed as a kind of pipelining through the various processing stages. If a process involves piping through eight stages these are now done sequentially; but if these stages could be piped to five different computers where the partial results of one were then fed to the others and acted as partial input for the next stage, then considerable time savings could be achieved. This may be the only way to get dramatic, orders of magnitude performance improvement.

MEMORY

The major trend in memory development seems to be toward lower and lower costs for ever larger memories. Indeed the trend might be said to be toward "zero cost" hardware solutions in memory and in mass storage devices. Nevertheless, at the present, memory costs remain significant.

A few years ago it appeared that laser read disks might offer a major breakthrough in this area. The present use of the laser read compact disks for storage, which now makes it possible to distribute very large geographic data bases to large sets of users by mailing each user a compact disk, is one present form of that technology. We can expect further developments in this area as the "writing" technology involved becomes more affordable to match the present low costs of the reading technology.

WORKSTATIONS

We are now seeing the gradual elimination of terminals in GIS hardware. They are being replaced by desk top computers. This means that for the price paid several years ago for a high resolution terminal, a high resolution screen together with compute power on the desk top can now be obtained.

By workstations, we mean interactive graphic workstations with very tight connections to the CPU so that integrated graphics and CPU power are available as a single tool, transparent to the user. One of the characteristics of the workstation is a very high graphic performance. Workstations add the ability to build much more sophisticated user interfaces than are possible with terminals by having the compute power and the high performance graphics on the desk top. Interactive graphic user interfaces, such as the MacIntosh or the Sun window/icon-based environments are possible. This will soon become a standard; people will expect this type of interface, which just isn't possible with terminals connected to central machines.

Workstations are of two kinds. One kind is the workstation that exists as part of a family of CPU's, like the VAX station and the high performance Sun or Apollo types. The other kind is the PC which is a workstation on a network. The latter often doesn't provide some of the networking and communication possibilities of the former; nevertheless, it's a very powerful standalone workstation that has some of the same fundamental components in it. Those workstations are becoming very fast. Upgrades to the PC/AT type systems (with something like the Intel 80386 chip running at 4 MIPS with a 32 bit word length) will make them equivalent to the MICROVAX in speed (and perhaps faster) for substantially lower costs. Moreover they can be linked with fast communication devices into larger networks.

What will be the impact of this kind of workstation? It will mean that processing, both analytic and graphic, can be done on desktop CPU's. So workstations and network technology will begin to have to work together in the GIS solution. One of the nodes on these networks still has to be a larger CPU for centralized processing and database management. The so-called file server device, which is a larger box shared by a number of workstations, has to be able to have information efficiently extracted from it, put onto local disks, and used locally, perhaps for updating, perhaps for analysis, perhaps for cartography, perhaps for reinsertion into a central library. Software solutions for organizing spatial libraries for rapid entry and extraction will work hand in hand with the hardware architecture of that central database. In our own software we have chosen the notion of spatial subdivisions in the form of tiles; these will allow pieces of the database to be pulled out into workstation environments for interactive work.

NETWORKS

In recent years the dominant hardware system architecture for GIS's has been the "multiuser host" system architecture; in this architecture one central processor supports a number of graphics and alphanumeric terminals. The single CPU provides computation, file and system management and drives the various shared peripheral devices (printers, plotters, etc.).

The network architecture is a rapidly emerging alternative to this multiuser host architecture. In this model there is a multiuser network rather than a single host. The network functions to integrate user workstations, compute servers, file servers, and shared peripherals. In such networks computing function is moved to single user workstations or to batch oriented compute servers. In its most extreme form each user has a dedicated workstation; less extreme networks have a combination of single user and multiuser nodes. The success of this approach is based on very fast networks making sharing of data practical and on inexpensive, high performance, single user workstations making the solution cost effective in comparison with the host oriented architecture. If such high performance workstations are available, the network is highly suited to the computation and graphics intensive applications of geoprocessing.

Hardware manufacturers are beginning to offer network architecture in their system configurations, meaning that their operating systems, their databases, and their communications, support the linking together of CPU's, and operate transparently to the end user. This means that different nodes within organizations have the ability to share and use data and software in common on a local area network. Wide area networks are also possible, making use of one of the various microwave transmission networks which are now being rapidly developing in the United States for the transmission of data over long distances. Hardware is thus being designed to support certain database concepts and this is very important for GIS. Instead of being just a set of tools, the hardware is becoming a fabric within which databases can easily be woven.

Also, because the networks usually use smaller CPU's as nodes, some of the degradation problems that we have had with one database being shared by many users are starting to go away; the hardware platform and architecture supplied by single vendors is becoming able to support a true database environment for geoprocessing.

A problem in using geographic databases in a network is that there are usually a lot of data to which the user needs access and there is also a lot of computational power needed on the desk top. PC's can give you the computational power on the desk top but you cannot share the data. A minicomputer can give you shared access to the data, but then you are competing for the same CPU with other people. The promise of networking is that it gives you the shared access and also gives you the performance at your desk top that you need for complex processing tasks and graphics.

We are just beginning to see the emergence of network architecture for CPU's by vendors such as Digital, Prime, Data General, Sun, Apollo and IBM. There are communication tools like Ethernet, that allow us to move among different vendors using TCP/IP, but we don't really see that successfully implemented at this time. Yet this is a trend that will eventually lead to multiple vendor hardware devices being connected in single networks. This will lead to both the inter-system and the intra-system database sharing.

One likely result of this trend is that there will be reduction in the power of the data processing management structures now in place; data processing managers and the whole computer center mentality will receive less emphasis and may eventually disappear. Instead transactional structures will be set up for updating, maintaining, managing and providing user access to the data.

HARDWARE FOR SPECIALIZED PROCESSING FUNCTIONS

One other trend is the development of hardware for specialized functions, especially relevant as movement toward networks occurs. Right now, in the network architectures, there are compute servers and file servers that do certain specialized functions for the network. There will be more development of other types of servers; for example, we already see sort servers and search servers such as the TRW Fast Data Finder and the Excel Sorting Engine. Other companies will be developing hardware assists for performance. Intergraph has done this a lot with their scanning graphics processors, and there will be an increasing trend toward moving geoprocessing into hardware to get increased performance.

In time, polygon overlay processing by intersecting tools may be built into a chip and become an integral part of geoprocessing machines. But there will have to be a much larger potential market for such chips before this occurs.

COMMENTS ON THE UNIX OPERATING SYSTEM

Despite some indications a few years ago that it might take over the operations system world, UNIX has not done so. Manufacturer's proprietary operating systems are still very strong and will continue to be supported indefinitely. Digital, with the introduction of their VAX cluster, local area cluster and the introduction of all their workstations that run VMS, is clearly committed to keeping VMS its operating system, or one of its two operating systems. IBM, with the introduction of minicomputers that run MBS and CMS, also have decided that they will continue to develop and support their proprietary operating system. What this means is that there will continue to be a diversity of operating system software and hardware and the differences between manufacturers will continue, thus making it more difficult to develop truly affordable software.

It will also continue to be difficult to connect different hardware devices in one big network. UNIX is not really an answer for making it all come together. Nevertheless, more and more boxes and workstations are committed to having UNIX as their only operating system.

Can the same kind of performance be obtained with UNIX operating systems by people like Sun or Apollo or others, that can be had from VMS or Primos operating systems? Probably these larger companies, because of their larger installed base, their larger proprietary interest, will devise new features in their operating systems that UNIX will not be able to keep up with. There will be attractive, end user-oriented types of solutions that UNIX just doesn't offer. The result will be two classes of machines. The inexpensive ones, based on operating systems like UNIX, will be fairly functional but won't do some of the really effective operations that proprietary operating systems will allow and offer.

One of the challenges for people who want to develop software that is machine independent will be to be able to develop very fast import/export procedures for traveling across machine environments with their data and also developing hardware which can run with standard compilers within different operating systems, with the operating system independent. One of the main things that we have done in ARC/INFO is the AML system which has freed us from some of the machine dependency on the command language processors which are resident in the operating system of each manufacturer; AML allows us to not only move across different vendor machines, but allows us to build macros which can move across them more effectively and pick up some of the key features from each of the different operating systems.

COMMENTS ON TRENDS IN PERIPHERAL DEVICES

The big trend in peripherals is the emergence of low cost reliable raster output devices. There are really two technologies now that are replacing the pen plotting type technology that we have used in the past. One is the electrostatic output, which is available in black and white and color. The other is the laser printer technology. The raster output device has great promise for increasing the quality of cartographic products because with the raster devices and the hardware that has been developed for these devices, half-tone patterns and high quality text fonts, high quality line symbols, and high quality point symbology are all possible and can be put directly onto the output media rather than having to be stroked in by pen. The other impact with the raster devices is that they can serve as input to pre-press page composition and graphic systems. Right now, there are only very expensive color, high resolution pre-press systems like Scitex's and inexpensive black and white small format systems like the Pagemaker on the MacIntosh. In time these two will converge and raster output from cartographic systems will go directly into high performance, color, pre-press systems.

A year or two ago scanners were being proposed as the new solution for input of cartographic data. While scanning has found its niche, it certainly hasn't taken over in cartography. Probably the main reason is that in a lot of data situations in cartography there is revision going on at the same time that manuscripts are being captured. The scanning technology still is not intelligent enough to deal with symbolized lines and many of the complicated text recognition problems.

NEW SPECIALIZED WORKSTATIONS

One of the devices that is badly needed for data entry is an instrument that allows data to be corrected as part of the data entry process. It will probably be a large, flat, display which will allow multiple planes of graphic memory to be used to display different layers of maps and images. It will allow compilation and adjustment to occur interactively among and between these layers relative to reference maps, base maps or images. What is needed next is hardware which can integrate display technology for pictures and vector screens and also serve as a platform for data entry and for overlay of manual maps and images. ESRI has performed manual map integration for about 15 years, using light tables and highly skilled craftsmen that know the relationships among various geographic data coverage types. They often make scientifically based decisions which result in cartographic adjustment to the data. With the introduction of this type of compilation tools, some kind of an interactive light table display mechanism with digitizer capabilities would be very powerful. We have also talked about using this instrument like an electronic sandbox to do thematic displays and suitability modeling in support of land use planning.

A host of new specialized workstations will also emerge. The analytic photogrammetric workstation is going to emerge into more and more of an integrated sort of device. The first ones were introduced this last year and they are still getting the bugs out, but some provide graphic super-imposition of database on the photogrammetric workstation.

Historically the whole orientation for workstation technology has been data capture, data editing, and some limited types of data use. There will be a trend in the next decade toward user workstations-- zoning workstations, land use planning workstations, water resources simulation workstations-- that actually are equipped with the kinds of output devices and analytic tools that allow users to do more than simply enter, display or edit data.

Human factors analysis has to enter into designing new kinds of workstations. One can envision the emergence of some kind of facilities, even entire rooms, that serve specific functions. Strategic planning rooms, emergency preparedness rooms, dispatch rooms, satellite tracking rooms, war rooms all have been equipped in the past with very crude ties to actual databases. Soon geoprocessing databases will be pipelined right into these rooms. Imagine full-wall CRT's which are touch-sensitive, panning and zooming on such a CRT, talking with it and having various tools to work directly with the data base. Rooms where users do geoprocessing will change. Users will not be sitting at a little CRT. Instead of just single persons, groups of people will be able to interact with the data; there will be informed conversations about situations; there will be more participatory planning.

CONCLUSIONS

The trend of rapid developments in hardware is continuing and these developments will have important effects in the next few years on the way in which geoprocessing is done and the way in which geographic information systems are structured.

THE MISGUIDED EVOLUTION OF FUTURE MAPPING TECHNOLOGY

David G. Gallant
Intergraph Corporation
One Madison Industrial Park
Huntsville, Alabama 35807
Tel.(205) 772-2000

ABSTRACT

Efficient fully automated mapping systems are impossible to achieve with current technology. As we move towards a higher degree of automation and sophistication, there exists an underlying assumption that faster and more accurate automated computer mapping systems will be the primary components of future cartographic production. So far the design approaches taken to develop these production tools have not fully considered the human interaction necessary for efficient map production, nor sufficiently addressed the automation problems of map production. In this context the components of a conceptual automated mapping system, aimed at including the human factor to support efficient cartographic production, are explored.

INTRODUCTION

The development of a computer system that will generate maps should be designed with the expected level of computer expertise of the systems' user factored into the design from the beginning. The integration of the human cartographer in the architecture of automated map production systems that exist today, would have made them more productive in less time. Compounding this inherent flaw is a trend to task persons such as electrical engineers, mathematicians and computer scientists with little or no formal training or experience in cartography or imagery analysis to develop functional map production systems. A little knowledge can be a dangerous thing.

I don't intend to rehash the previous discussions of 'man vs machine', or evaluate the same questions concerning what part 'man' should play in the computer mapping process. Instead, I will offer an approach to automated mapping that I consider to be unique and innovative for the discipline of cartography and industry of computer mapping. My approach includes the active involvement of human cartographers throughout all phases of the automated mapping process.

To reach the stated goal of this paper, I will briefly trace the evolutionary path of automated map production systems. Following this background discussion there will be a look at the current state of automated mapping systems and the major flaws inherent in their architecture. Finally, future directions are explored through the presentation of a conceptual automated mapping system for current and future needs.

EMERGENCE AND DEVELOPMENT OF COMPUTER MAPPING TECHNOLOGIES

Tasking computers to generate and manipulate spatial information for mapping purposes is a process that has evolved rapidly since the 1960s. In earlier phases of computer mapping much of the production process was done in batch mode and focused on numerical analysis. Output was in the form of graphical representations such as bar graphs and charts which attempted to portray a particular mapping theme. Automation of cartographic processes and methodologies to portray spatial information in map form were slow and time consuming. Also, final graphical representations of mapped information were crude and simplistic.

In the mid 1960s to early 1970s there emerged an effort to develop new methods and technologies which would offer a more productive alternative to early automated mapping processes. Electronic displays for graphics improved, and minicomputers made mapping systems more economical. Some of the first functional computer mapping systems with an interactive component/capability emerged during the 1970's.

By the early 1970s imagery from orbiting space satellites emerged as a possible major source of information for map generation. Such satellites as the Landsat series beginning in 1972, and space programs such as Skylab, Gemini, and Apollo helped propel the discipline of cartography into a new age of digital awareness.

Despite the problems inherent in the development of computer mapping systems such as map accuracy, scale change, orientation, data sizing, etc., work continued towards development of faster and more automated methods of map generation. As a result, a large part of mapping hardware and software development over the past several years has focussed on 'faster' methods of trying to store, access, manipulate, and process large amounts of digital cartographic information. These developments helped set the stage for the misguided evolution of future developments in cartography by placing less and less emphasis on human cartographic knowledge and skills.

The use of computer graphics to simulate map characteristics and display results on a terminal screen, eventually evolved into an art form of its' own, and along with this new art form came the overlooked and as yet unanswered questions of digital standardization. Questions about digital cartographic issues involving map symbolization, color, size, style, etc., were far more numerous than answers and as varied as the computer graphics organizations themselves.

Fortunately efforts like the Federal Interagency Coordinating Committee on Digital Cartography (FICCDIC), and the National Committee for Digital Cartographic Data Standards (NCDCDS), are attempting to sort out and place in perspective some issues regarding digital cartographic standards. This type of digital standards research holds a great promise for approaching excellence in processing digital data through a cartographic computer system, and turning it into useful map products

CURRENT APPROACHES

In order for some automated mapping organizations to maintain a high productivity, and an economic return on their investments, there is a tendency to have staff specialized on one instrument of the mapping process (BIE 1984).

A similar approach is used in development of mapping software, whereby different cartographic operations are segregated and tasked to a particular software designer to package. Thus we have got the names placement package, the symbology package, the generalization package, and so on. The synthesized skills of a human cartographer are modularized for computer imitation and the resultant mapping system is not unlike a robot-run automobile assembly line. However, with human specialization there is a decrease in knowledge and skills that are exchanged between stages in a map production process (BIE 1984).

Therefore, computer mapping organizations should be considering, to what degree loss of cartographic knowledge and skills are accumulating throughout the computer mapping process, since rarely do the same persons, or persons trained in cartography, develop the assortment of mapping packages that are combined to make one mapping system. And although various packages are tested for functional compatibility, there doesn't appear to be any testing of the viability of the map products it can turn out, which is a test of the effectiveness of any cartographic production system.

As the design of automated map production systems has matured and evolved, a variety of techniques such as artificial intelligence, automatic pattern recognition, and syntactical analysis, currently being used in the mapping industry have been implemented Unfortunately automated mapping techniques conceptualized as a tool to multiply human cartographic capability, are now evolving into tools that are independent of humans. The developers of expert systems such as ACES, which is "an attempt to capture the expertise cartographers use in the labeling process", concluded that much of the map labeling process can be automated, and to them it was clear that many other cartographic tasks could be applicable to artificial intelligence techniques. (Pfefferkorn, Burr, Harrison, Heckman, Oresky, Rothermel, 1984). ACES is a prime example of the potential misuse of technology.

FUTURE DIRECTIONS

Major developments are already under way in areas of knowledge based systems and parallel processing hardware. These directions will likely continue into the future. However, in order to achieve an efficient automated mapping system designed to produce accurate and effective maps, a new direction for computer mapping is needed in which the unique human cartographic capabilities are identified explicitly and then fully integrated into the overall system design. A good place to start in designing an efficient automated mapping system is to maintain and consult cartographic knowledge gained over past centuries, and seriously consider the question/goal of whether a fully automated mapping system is a good idea that represents a true advance in cartographic production. An overview of the conceptual system I am recommending is given in the following paragraphs.

The first step towards design and implementation of this conceptual system, is to require a cartographer to berome a major part of the initial design and implementation team. This will ensure that software packages and commands used to generate a map be compatible with accepted cartographic practices (Figure 1).

The next sequence of steps will require a cartographer working with an image analyst, to evaluate hardcopy digitized/scanned material and digital satellite imagery for use as base information, prior to being placed in external storage, or made part of the computer files (Figure 2).

Completing the previous tasks, the manipulation process for further reviewing, correcting, and enhancing cartographic data must be performed by a cartographer In the case where satellite imagery is used as a basis for creating maps, the cartographer would work along side of an image analyst At this level of the map design the image analyst would serve as a consultant in interpreting imagery data with relation to spectral signatures, relief displacement, vertical exaggeration, and multiband comparisons of imagery information which could be used as a spatial information foundation from which the cartographer could build a map.

It is at this level of the system that a cartographer will have most interaction with the system when compiling information for a map. Here the operations of a mapping system fall into two categories, functions that act upon data without change, and functions that act upon data with change. However, a cartographer sitting at a terminal can halt, or change any operations as they occur. It is these functions that will require most development in the future if we are to allow for maximum human interaction (Figure 3, upper left corner).

Once a desired data combination is reached the file is then sent to the 'automatic' part of the mapping process. This step is envisioned as merging multiple planes of information into one plane (Figure 3, lower left corner). After this is done the suggested output product configuration is evaluated by the cartographer for acceptance or rejection. This would ensure that the output product contains appropriate information for use by the intended user (Figure 3, right side).

CONCLUSIONS

The evolution of computer mapping systems has benefitted the practice of cartography by providing better tools for the more rapid display and manipulation of compiled information. Additional work to improve this capability should be continued. However, automated mapping system architecture lacks the basic experienced based knowledge (know-how) necessary to identify, evaluate, and correct badly designed maps produced by automated processing. Currently, only an experienced cartographer can make the final decision, judgments about whether a particular arrangement of symbolized graphic data in map form will effectively communicate to the products' users Therefore, future developments in mapping systems should be centered around using the computer as a cartographers tool. The focus should not be a system as a substitute for heuristic cartographic knowledge gained only through experience; until such time as an efficient methodology for the incorporation of heuristic knowledge is achieved.

It is unfair to blame the computer industry for all of the misguided directions of computer mapping development. After all, if there were enough qualified cartographers to participate in the development of automated mapping systems (and they were allowed to do so) some of the misuse I pointed out would not occur. As (Taylor 1984) states unless cartographers get involved in the 'New Cartography' they will be replaced by computer graphic designers that don't possess the adequate background or knowledge of basic map design techniques.

The future skill level, and training of the cartographer must include experience with a programming language other than BASIC or FORTRAN, and appropriate math skills such as trigonometry and calculus. These skills must be placed upon a firm foundation of the conceptual and communication fundamentals of mapping. The cartographer must also be skilled at some level in image processing, and future skill levels of image analysts must include advanced skills in cartography. Otherwise, as we approach an environment where digital satellite imagery is used as a primary source of information to produce hardcopy maps, Image Analysts could find themselves heading down the same path of non-involvement as Cartographers, when it comes to automated mapping system design.

To achieve this background training and knowledge, more institutions of higher learning are going to have to offer course loads directed at imagery analysis and processing techniques, as well as courses centered on traditional and computerized methods of cartography. Without this breadth of effort to formally educate more people in cartography and imagery analysis, and without changes in the computer mapping industry to allow for more human expert interaction, the misguided evolution of future mapping technology will continue.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Alan A. DeLucia for his valuable suggestions and help in preparing this paper.

REFERENCES

Brassel Kurt, 1977, A Survey of Cartographic Display Software, International Yearbook of Cartography, pp 60-77

Bie, Stein W., 1984, Organizational Needs For Technological Advancement, Cartographica-AutoCarto Six Selected Papers, Vol. 21, No. 2 & 3, David H. Douglas, ed.

Burr, David J., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Harrison David A., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Heckman Bradford K., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Oresky, C., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Pfefferkorn, Charles E., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Rothermel John G., 1985, ACES: A Cartographic Expert System, Auto-Carto 7 Proceedings, pp 399-404, Falls Church, American Society of Photogrammetry and American Congress on Surveying and Mapping.

Taylor D.R.F., 1984, New Cartography, 12th International Conference, International Cartographic Association, Vol I, pp 455-467, Perth.

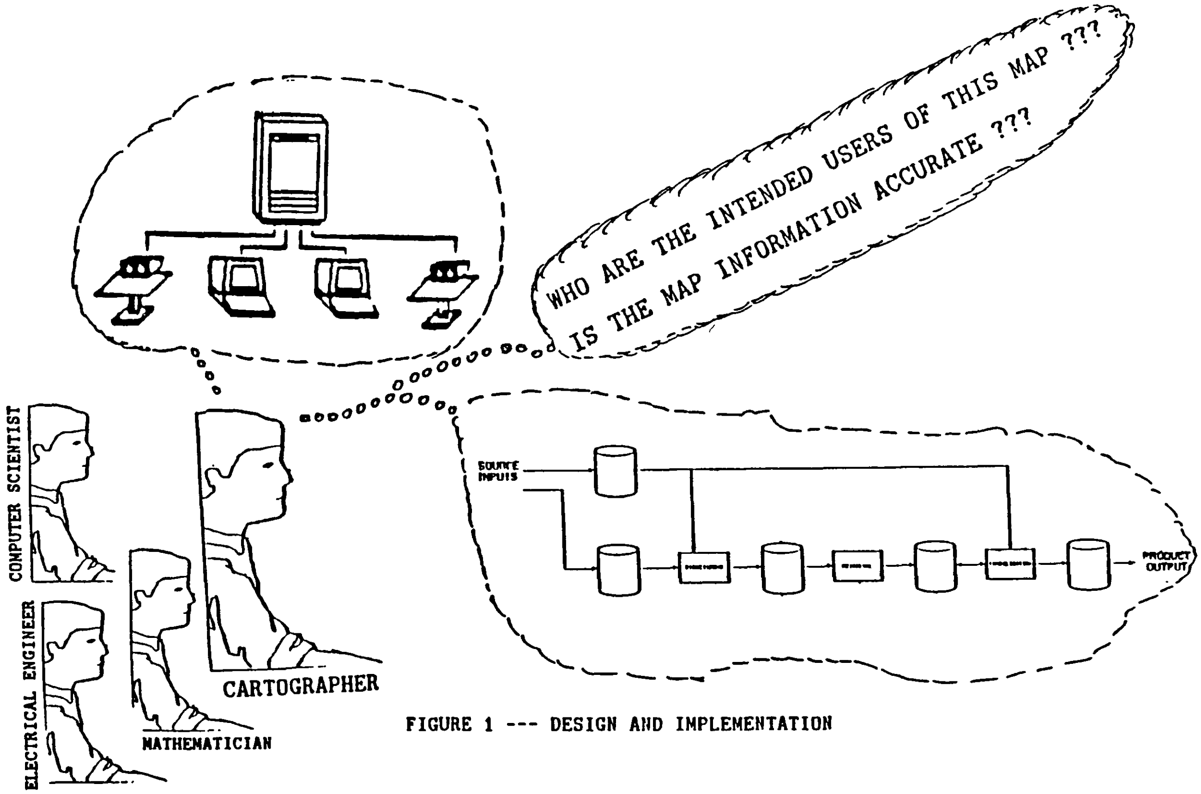


FIGURE 1 --- DESIGN AND IMPLEMENTATION

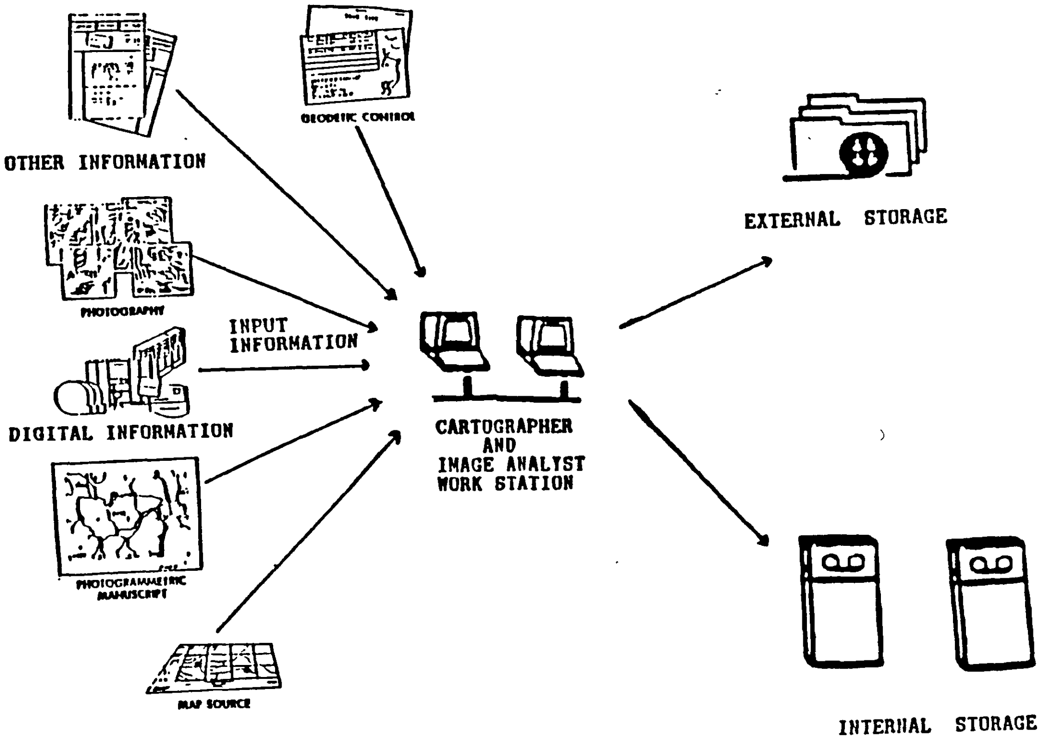


FIGURE 2 --- EVALUATE INCOMING MATERIAL

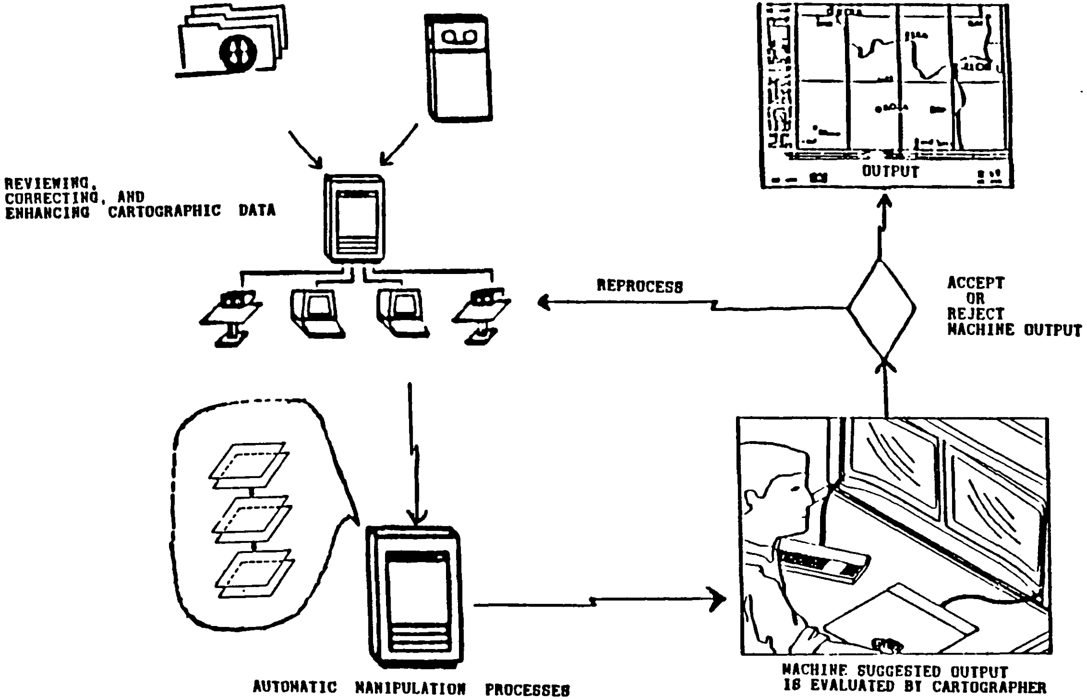

REVIEWING, CORRECTING, AND ENHANCING CARTOGRAPHIC DATA
REPROCESS
OUTPUT
ACCEPT OR REJECT MACHINE OUTPUT
AUTOMATIC MANIPULATION PROCESSES
MACHINE SUGGESTED OUTPUT IS EVALUATED BY CARTOGRAPHER

FIGURE 3 --- PROCESSING STEPS

HUMAN INTERFACE REQUIREMENTS FOR VEHICLE NAVIGATION AIDS

Matthew McGranaghan
Department of Geography
University of Hawaii at Manoa
2424 Maile Way
Honolulu, HI 96822

BIOGRAPHICAL SKETCH

Matthew McGranaghan earned BA and MA degrees at SUNY-Albany and a PhD in Geography at SUNY-Buffalo. He is an Assistant Professor of Geography at the University of Hawaii at Manoa. His research interests include color perception in map displays and human factors considerations in map design.

ABSTRACT

The development of vehicle navigation aids has been facilitated by advances in microprocessor, display, and data storage technologies. While many of the problems associated with creating these devices have fallen to technological developments, several questions remain. These arise when people must use the devices. Navigation aids must be designed with the human user and the driving environment in mind. As the basic technology of processing navigation information comes to maturity, the remaining questions depend for answers on understanding human cognitive and perceptual parameters of the navigation task.

INTRODUCTION

Navigation aids for drivers should provide support, in the form of information, which will assist drivers in traversing geographic space. In addition to informing the driver of the vehicle's current position, usually with reference to the street grid and on a map display, these systems may display trip origins, destinations, and pre-selected or optimal routes. Other information, such as the location of the nearest hospital, police station or hotel could be found, given a properly constructed database. All of these functions have been provided, in the past, by a human navigator using printed street maps.

Microprocessor-based electronic navigation aids, are capable of storing and displaying map information. The advantage of a vehicle navigation appliance (VNA) over a printed map lies in the additional capabilities the former may offer beyond the printed map. Computing technology allows navigation aids to include map-use and navigation expertise as well as map information in the system. Such tasks as: route selection, route optimization, continuous real-time position tracking, and deciding on the next action to be taken in following a route, can be performed by the navigation aid. This potentially frees the driver from at least some of the effort required for navigation. Resulting benefits may include safer, more efficient, and less stressful travel.

We may conceptually divide a VNA into two functional components: an underlying spatial data processing system, and a human interface. Data structures, storage medium, positioning system, and the content of the spatial database will be included in the underlying system. The interface is a facade, through which the driver interacts with the underlying system. Both of these components depend on software and hardware subsystems to support their functions. In practice, the two components are interdependent.

It is reasonable to expect that the basic hardware technologies and data processing techniques to support the spatial data processing components of different brands of commercially available VNAs will be comparable. Different information bases may be used for different applications (e.g. tourists, salesmen, and firemen may each have databases suited to their specific needs). Oil companies, fast food concerns, hotel chains and the like may well vie to have their franchises incorporated into databases. It seems likely that very similar databases will be available for VNAs produced by different manufacturers. The area where VNAs will be most free to differ, and therefor, to attract users (and garner market share) will be the interface they present to users.

The remainder of this paper deals with the interface component of vehicle navigation aids. It considers the technological basis available for the interface and concentrates on the creation of a useful and supportive navigation aid. This area presents interesting challenges for cartographers in the future development of VNAs.

THE INTERFACE

Function

The function of the human interface of a VNA is to facilitate communication between the VNA's underlying processing component and the driver. This will entail information flowing both ways; the driver must be able to control (to some extent) the VNA's processing and the VNA must be able to present information to the driver. These flows should be comfortable and natural. It is clear that the information should be provided in a useful format and should be tailored to the driving environment. The style of communication must not interfere with the driving task.

Input to the VNA, (such as queries for particular information, parameters to control a search, or commands to abandon a task) present challenges which will only be mentioned here. Obvious technologies include keyboards, pointing devices, and voice activation. Each has some associated disadvantage. Keyboards require space and are not usable with the vehicle underway. Light-pens would be a nuisance; while capacitance-based touch screens (and keyboards) would not work with gloved hands. Pressure sensitive touch screens are appealing but accurate pointing/touching may be difficult in a moving vehicle; and made more so by small target areas.

The output from the VNA must meet several requirements. The first is that the information must be perceptible to the driver. Second, the information must be useful, i.e. support the driver. Third, the information must be presented in such a way that the driver need not spend undue effort decoding the message.

The first requirement deals primarily with matching perceptual abilities of the human visual system with available display technologies. Concern for sufficient symbol size and contrast to ensure visibility is indicated. This type of basic system parameter is relatively well understood (see, for example, Silverstein 1982).

The second requirement, that the information is useful, is less well understood. It is task dependent and undoubtedly unstable through time. It is very difficult to presage all of the information that may be of use to the driver. Toyota has considered on-line automobile owner's manuals, automobile condition monitoring, real-time traffic reports, navigation information and commercial television broadcasts as information worthy of display (Shoji 1986).

The third area is concerned with the form, as opposed to the content, of the information presentation. It is probably the least understood of the three requirements. Meeting this requirement will entail an understanding of cognitive aspects of direction giving and receiving. It is relatively simple to create a VNA interface which looks like a conventional street map. The vexing question is how to use the technological power to go beyond the limits of that map. It is difficult to envision new forms of spatial information displays based not on the limits of static printed maps but rather on the palette that current technology provides.

Cognitive Load in the Driving Environment

Driving is a highly visual activity (Hughs and Cole, 1986). It requires a driver to use a large amount of sensory information to constantly assess the current condition of the vehicle and its situation with respect to the road and other traffic. Further, knowledge of the current situation, a route to be followed, the operation of the vehicle, and applicable traffic laws and conventions must be combined to decide the next action the driver will take.

Psychologists have indicated that there are limits to how much information a person can attend. The processing demands on the driver are such that "... 1 in 10 traffic signs are noticed" (Hughs and Cole 1986 p. 389). Still, those authors found that between 30 and 50% of the attention of drivers was being allocated to non-driving-related objects (p. 388).

Perceptual Requirements

Independent of the form or content of a message from the VNA to the driver, it is crucial that the driver be able to pick the message out of the sensory noise found in the driving environment. The message must be sufficiently conspicuous to gain the driver's attention, be processed, and finally, used to support navigation. The human visual system is of most interest for transmitting navigation information. The performance of this system is effected by the amount of non-informative stimulation received by the sensor.

The passenger compartment of an automobile is subject to fluctuating levels of lighting. The illuminance in an automobile may fluctuate over a wide range. Bailey (1982 p. 58) estimates this range to be from 0.0001 millilamberts in a garage or under starlight to 10,000 millilamberts under the midday sun. The human visual system exhibits adaptation to the level of illumination. This adaptation impacts the visual system's ability to detect small symbols (Murch 1985 p. 2).

Carter and Carter (1981) indicate that high conspicuity increases the likelihood of any particular piece of information being attended. Visually encoded VNA information must be visible i.e., have sufficient conspicuity to facilitate its use. Symbol conspicuity is a function of the contrast ratio in the display. Acceptable contrast ratios range from 3:1 through 50:1 with 10:1 being optimal (Murch 1985 p. 3). Murch (p. 5) indicates that contrast ratios of 3:1 can be maintained on avionics CRTs with up to 6500 footcandles ambient illumination while maintaining acceptable display luminance levels. Shoji (1986) indicates that CRTs are also useful under very low levels of illumination.

Competing Technologies

It is possible to present the information on either a directly viewed device or to project it on the vehicle's windshield, a heads up display (HUD). The systems currently available in this country rely on directly viewed display screens mounted within the driver's compartment. Mitsubishi Motors Corporation has tested a LCD-based prototype HUD for passenger car use (Horikiri _et al_. 1986).

Several display technologies could be used to for directly viewed display devices. The most likely are CRT, LCD and LED devices. While all three could support graphic displays, the rugedness, range of luminosity, resolution, and color capability of the CRT indicate it will be predominant for some time (Bailey 1982). The required mounting depth for CRTs limits their positioning within the driver's compartment, but CRT depths are being reduced (Shoji 1986).

Information Requirements

The information actually required to guide a driver through a street network is quite limited. Research, by Benjamin Kuipers (1978), by Streeter (1985, _et al_. 1985) and by Riesbeck (1980), suggests that, at each point in the navigation task, it would be sufficient to indicate to the driver: 1) what action to take next; and 2) when to take it. The overview perspective inherent in a conventional map is not needed to follow a route. Assume that the processing component of the VNA has already selected (or been provided) a route to follow, and that this route has been decomposed into actions to take at recognizable places (Kuipers' view-action pairs). Some sort of warning of an impending action (turn left or right) and an indication of the place to take the action would be adequate. A VNA with this ability would be as handy as a passenger familiar with the route telling the driver the same information. This approach to direction giving closely mimics the way human route expertise is often provided: as needed. The VNA's utility would be greater if it could recognize deviations from a planned route and generate a new sequence of instructions leading from the current position to the destination, with the same facility as a human expert.

Other potentially useful information will vary widely for specific uses of VNAs. The range of this information was indicated above. The result of this diversity of desires may be a proliferation of special purpose databases capable of supporting user's queries. The commercial value to service industries of being listed in "drivers' GISs" is considerable. It alone is apt to ensure that third party venders will supply customized databases for VNAs.

Form of Presentation

The form of information presentation used by a VNA is at the heart of the human interface. This discussion concentrates on using the human visual system (as opposed to the auditory or tactile systems) to give information to the driver. There are several ways in which spatial navigation data might be visually presented. These can be ordered along a contiuum of abstraction from the underlying geographic reality. Different levels of abstraction may require different amounts of effort for the driver to understand the message. One concern is to maintain stimulus-response compatibility. The message to the driver can, by its form, suggest the next action to be taken.

At the most abstract end of the continuum, spatial information could be encoded as verbally, as words, on an alphanumeric display. Instructions such as 'Turn left at the next intersection' could describe navigation procedures. The presentation would be essentially aspatial. This approach requires relatively low display technology but a high degree of sophistication in the spatial data handling subsystem. Reading of alphanumerics is better understood than reading spatially structured (map) displays.

A more pictorially oriented display could be used to indicate the next action to be taken. The Nissan Cue-X HUD system provides an arrow pointing in the direction the driver should turn (Okabayashi and Chiba 1986). This form of message should exhibit good stimulus-response compatibility. Symbols indicating 'prepare to turn left' or 'use the left lane' might be color coded versions of the stronger 'turn left here' type of message. An 'on course' indicator could provide comforting feedback to the driver.

Further along the continuum, a display might provide a simplified schematic view of the route to be followed. Such a custom map might dispense with streets which are not part of the route to be followed. The topological structure of the route, may be emphasized over the planimetry. A position indicating symbol would desirable (perhaps necessary) in such a display. This type of display presents more information than the driver needs at any instant for route following. Subsequently, higher processing demands are placed on the driver. The display device would have to be flexible and support graphics output. The processing subsystem could be relatively unsophisticated; once it has found a route, it need only map stored coordinate data for each of the street segments to be traversed into display device coordinates.

Planimetrically correct map displays of complete street networks are the least abstract display class. They closely mimic the geographic reality, consequently they may require the driver to do more of the navigation work - potentially as much as would a printed street map. The processing capability required is fairly low; display a 'you are here' symbol, possibly a 'destination' and the street segments from the database. There are, of course, questions of scale and area covered, tracking style (north-up vs. heading-up, move the 'you are here' symbol on the street network or the street network under the symbol) and map design (e.g., the use of color, symbol selection, and inclusion and placement of street and place names).

CONCLUSION

My expectation is that the VNA which presents the simplest appearance to the driver will, in the long run, be the most successful. It may also be difficult to market because of its simple appearance. The appearance will belie, and be inversely related to, the underlying data processing system. Both processing capability and database will be highly sophisticated and extremely reliable. The driver will follow simple instructions trusting them to be correct. The VNA must recognize deviations from its instructions and take them in stride.

REFERENCES

Bailey, R., 1982, Human Performance Engineering: A Guide for System Designers, Prentice-Hall, Englewood Cliffs, New Jersey.

Carroll, A., 1986, Touch technology: variety of systems spur maturity, Information Display, v.2, n. 11, pp.15-20.

Carter, E., and Carter,R., 1981, Color and Conspicuousness, Journal of the Optical Society of America, v. 71, n. 6, pp. 723-729.

Cooke, D., Vehicle navigation appliances, Proceedings, Auto-Carto 7, pp. 108-115, 1985.

Horikiri, K., Ueda, F., Kumagai, N., and Saga, I., 1986, A Head-Up Display for Automotive Use, Proceedings of the 6th International Display Research Conference, September 30 - October 2, 1986, Tokyo.

Hughes, P., and Cole, B., 1986, What Attracts Attention When Driving?, Ergonomics, vol. 29, no. 3, pp. 377-391.

Kuipers, B., 1978, Modeling spatial knowledge, *Cognitive Science* 2:129-153.

Mark, D., 1985, Finding simple routes: 'Ease of description' as an objective function in automated route selection, *Proceedings, Second Conference on Artificial Intelligence Applications (IEEE)*, Miami Beach, December 11-13, 1985, pp 577-581.

Mark, D., and McGranaghan, M., 1986, Effective provision of navigation assistance to drivers: A cognitive science approach. *Proceedings, Auto Carto London*, v. 2, pp. 399-408.

Murch, G., 1985, Visual Demands for Avionics Displays, Tektronics, Beaverton OR.

Okabayashi, S., and Chiba, M., 1986, Recent Trend in Electronic Displays for Automobiles, *Proceedings of the 6th International Display Research Conference*, September 30 - October 2, 1986, Tokyo.

Riesbeck, C., 'You can't miss it': Judging the clarity of directions, *Cognitive Science* 4:285-303, 1980.

Shoji, Y., 1986, Development of a CRT Display System for an Automobile, *Proceedings of the 6th International Display Research Conference*, September 30 - October 2, 1986, Tokyo.

Silverstein, L., 1982, Human Factors for Color CRT Displays.

Streeter, L., 1985, Comparing navigation aids for computer-assisted navigation. *Paper Presented at Auto Carto 7* (not published in proceedings).

Streeter, L., Vitello, D., and Wonsiewicz, S. A. 1985. How to tell people where to go: Comparing navigational aids. *International Journal of Man/Machine Studies* 22, 549-462.

White, M., Building a digital map of the nation for automated vehicle navigation, *Proceedings, Auto-Carto 7*, p. 570 (abstract only), 1985.

Wiener, E. L., 1985, Beyond the Sterile Cockpit, *Human Factors*, 27(1), pp.75-90.

Wong, K. W., and Yacoumelos, N. G., 1973, Identification of Cartographic Symbols from TV Displays, *Human Factors*, 15(1), pp. 21-31.

ENHANCEMENT AND TESTING OF A MICROCOMPUTER-BASED GIS FOR UNIVERSITY INSTRUCTION

Duane F. Marble
Sherry Amundson
Jatinder S. Sandhu
Geographic Information Systems Laboratory
State University of New York at Buffalo
Amherst, NY 14260

ABSTRACT

This paper traces the history of the Map Analysis Package and reviews the enhancements made to its user interface for university instruction use. Relationships between database size, available system resources and processing speed are discussed.

INTRODUCTION

University instruction in geographic information systems must be carried out in a laboratory environment where the student has extensive access to "hands-on" work with a GIS and extensive, real-world data sets. Major commercial systems are expensive to acquire and operate and are normally out of reach of most academic institutions. One answer to this problem is to utilize much less expensive software which retains the basic functional structure of the GIS at a level which permits the student to acquire a working knowledge of basic GIS operations.

The mainframe version of the Map Analysis Package (MAP) was created several years ago at Yale University by Dana Tomlin and Joe Berry. The program was widely distributed at a low price and has been used by many universities as a teaching tool in GIS. The FORTRAN source code was made freely available by Yale and, as a result, versions were ultimately available for most mainframe and minicomputers. However the mainframe version supported only character output on CRTs and line printers, documentation was limited, and no enhancements had been made since the original release of the software.

About two years ago MAP was rewritten in Pascal for the IBM PC and released as a commercial package. From the standpoint of university instruction this package (PMAP) had several severe disadvantages: it was much too expensive, the lower-cost instructional version (AMAP) was so limited as to be nearly useless, execution was very slow and the source code was not available for local examination and modification.

In 1985 Dr. Dana Tomlin (now at the Graduate School of Design, Harvard University) received support from IBM to translate the original FORTRAN source code to an IBM/AT environment. In his original, draft version much of the code was little changed from the older mainframe version, but some color graphic enhancements were added. Harvard has since released an improved version of this software, including the FORTRAN source code, for a low price and is continuing development of the program (Tomlin, 1986).

Late in 1985 Dr. Tomlin kindly made a copy of the rough draft program and its source code available to the major GIS programs at Buffalo, Penn State and South Carolina. These institutions had a significant and immediate need for a very low cost teaching tool in the GIS area. Examination of the initial draft of the code indicated that several significant problems needed to be corrected before the program could be effectively used in a university instructional environment.

Among the general problems were a very restrictive user interface, new graphic functions which made use of the IBM Virtual Device Interface (VDI) drivers and a number of functional limitations which had been carried forward from the older, mainframe version (e.g., lack of an operational HELP function within the program). Because of time pressures, the three institutions began independent testing and enhancement of the draft Harvard software early in 1986.

Development Activities at South Carolina and Penn State

South Carolina undertook the initial effort to improve the draft program by quickly creating a basic working version which was sent to Buffalo and Penn State early in 1986. Additional work by South Carolina substantially increased the size of the analysis area supported (from 10K cells to about 60K), redesigned the user interface, and provided greatly improved facilities for data input. Several functions (e.g., error counting) were removed and a basic HELP function was added (Rasche and Cowen, 1987). This version was made available to Buffalo in August of 1986 when South Carolina temporarily ceased further development work on the program to concentrate upon database creation and program testing.

Penn State also made a number of changes in the original South Carolina version to support their immediate teaching needs in the Spring of 1986 and has since concentrated upon development of exercises to support their instructional work in GIS.

Development Work at SUNY at Buffalo

While South Carolina had made substantial improvements in the original Harvard code, analysis of their Fall 1986 release indicated that the result was still not suitable for instructional use within our particular instructional environment. (A faculty-wide PC lab currently supporting only monochrome graphics and strict university rules about purchasing copies of all software for student use.)

Among the major problems that remained were a continued dependance upon the expensive ($499 list price) VDI drivers for graphics (which even with a substantial discount from IBM we could not afford to purchase in multiple copies for student use), screen designs that we felt still did not make maximum use of the limited, low resolution graphics displays, and very limited internal and external documentation. Good internal documentation of the program was essential since we also intended to use the source code as a laboratory tool in advanced GIS courses, as well as a direct teaching tool in the introductory GIS applications course.

From an instructional standpoint, the package also suffered from a lack of well organized and documented data sets which could form the basis for student exercises. Our instructional methodology requires the databases to be large and representative of real-world problem situations. Two well-documented data sets had been created for foreign areas (Panama and South Thailand), but more were needed for adequate instructional work.

A major technical problem was replacement of the VDI drivers. Nearly all device-independent graphics packages for the IBM PC, such as the HALO package from Media Cybernetics and the various versions of VDI and GKS, suffer from one major flaw from a university instructional standpoint - they must be paid for! An examination of the software marketplace revealed one exception in the Fall of 1986: MetaWINDOW created by the Metagraphics Software Corporation of Scotts Valley, CA.* MetaWINDOW provides a fairly rich function set and, in one version, operates through a memory-resident driver which may be distributed on a royalty-free basis with developed applications. It also supports a wide range of graphics displays, but currently no hardcopy plot devices. The graphics calls may also be made from either Fortran, Pascal or C.

PROGRAM ENHANCEMENTS

Conversion to MetaWINDOW

The South Carolina version of the MAP package used GSS*CGI library routines and required their device drivers to be memory resident. These drivers supported only the Enhanced Graphic Adapter and could not be distributed with the executable version of the program.

The first phase in the conversion from VDI to MetaWINDOW graphics involved translating the MAP program from IBM Professional Fortran (a full ANSI standard compiler), to Microsoft Fortran (a subset of the ANSI standards). In most cases, this was a straightforward process except where opening files, string concatenation and passing huge arrays (more than 64Kb) was concerned.

The second phase involved converting each VDI graphic function call to a MetaWINDOW graphic function call. Here again, most operations were one-for-one translations. In the case of the COLOR command (Rasche and Cowen, 1987), no single MetaWINDOW function could replace the VDI function

* MetaWINDOW is a registered trademark of the Metagraphics Software Corporation.

used. The command had to be rewritten entirely.

Enhancing the User Interface

A well designed user interface can reduce the "learning" time of a system, increase human processing speed and reduce operator errors. Desirable things to have in a student use environment are: computer assisted startup, on-line help for the user, high levels of program reliability (error trapping) and the use of consistent function keys for basic operator inputs.

MetaWINDOW has a variety of advanced features that assisted us in improving the user interface. Some of these are the ability to have multiple ports on screen, user-defined virtual coordinates and clipping limits for each port, ability to open windows and key hit detection. This allows each key hit to be parsed and handled within the program. It is also possible to use multiple fonts of various sizes on any port.

Initial use of the program has been made easier by having an on-line help facility at the touch of a function key. This is especially helpful to the first time user. All databases in the current directory are displayed. The user selects a database for processing by simply highlighting it using the cursor keys. Once the MAP program is initialized, a map of the function key assignments is available as a pull-down window.

The screen design of the South Carolina version of MAP divided the screen into a display area, a legend area and a command line area (Rasche and Cowen, 1987). In designing the new screen interface, the bottom 5% of the screen area has been reserved for entry of user commands, leaving the remainder fully available for graphic output. Any map image drawn on the screen is scaled to the screen dimensions and centered. It remains on the screen until a new graphic command is executed or the user requests the screen be cleared. This allows the user to refer to the map image while formulating and executing subsequent commands. In addition any changes to the palette are reflected instantly on the map image. The text display is restricted to a window located in the top part of the screen which can be toggled on or off by the user to see the image underneath (see Figure 1). The length of a single line of text display has been increased to make commands with tabular output more readable.

Help for individual commands is now available at two levels. When help is requested through the EXPLAIN function, a full description of the command is displayed in the text window and a syntax diagram is displayed in another window located just above the command line. Both windows may be toggled on or off with function keys. There is no restriction on the length of the description because the user may page through it; it is truly an on-line manual. In addition the user may request just the syntax diagram for reference, midway through typing a command.

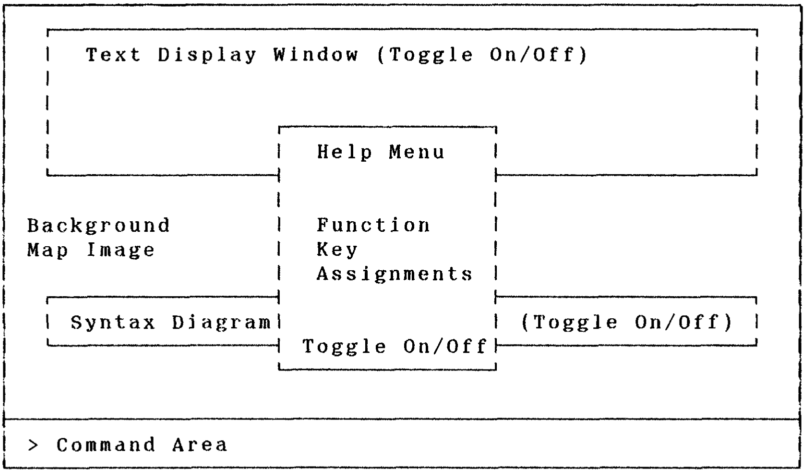


Figure 1

Figure 1 shows the position of the text display window, the syntax diagram window and the help window when they are pulled down over the map image area. A cursor has been added in the command area, and pressing the ESC key clears the command line altogether. These features, in addition to the fact that any of the windows will toggle on and off at any time, facilitate user entry of commands.

The display commands COLOR, CONTOUR, and SURFACE have been made faster by run encoding. On a plotter this would reduce the large number of pen up and pen down movements, a problem encountered in the South Carolina version of the program (Rasche and Cowen, 1987). An option has been added to the color command to represent map categories with patterns instead of colors. The legend has been redesigned to improve map readability by displaying only the categories actually shown on the map.

Hardware Independence

The MetaWINDOW package provides functions that can detect the particular graphic hardware installed and its capabilities, and it provides run time support for a number of popular boards such as the EGA, CGA, and Hercules Monochrome Graphics. The Buffalo version of MAP for the PC has been written to detect these various configurations and adjust itself at runtime, without the user having to worry about what particular graphic device is present. The COLOR command, using the pattern option, will work even on monochrome graphic displays.

Some Limitations of the MetaWINDOW Conversion

Hardcopy output is restricted because MetaWINDOW currently supports only two hardcopy devices: an IBM graphics or

Epson compatible dot matrix printer and a jLaser laser printer interface. Also displaying the text in windows without erasing the background image requires that a substantial amount of memory be set aside for saving screen images.

PERFORMANCE EVALUATION AND TESTING

The Buffalo version of MAP for the PC runs on a minimum hardware configuration of an IBM PC/XT or AT or close equivalent, with 640K of memory, an optional 8087/80287 math co-processor, and a 10MB hard disk. Graphic output may be displayed on any device supported by MetaWINDOW, such as the Enhanced Graphics Adaptor (640 x 350 resolution), the IBM Color Graphics Adaptor (320 x 200 resolution), or the Hercules monochrome graphics adaptor (720 x 348 resolution).

The program allows the user to inadvertently initialize a database that exceeds available disk space. Therefore we have tested the software to identify the limits to data volume imposed by the system, and the level of disk storage required by different data volumes. In addition we have conducted various performance analysis tests to establish tradeoffs between data volume and processing speed, and between data volume and display resolution. The full results of these tests are available in a technical report released by the SUNY at Buffalo GIS Laboratory (Amundson, 1987). The report provides users with a guideline for setting limits on data volumes and for matching hardware configurations to their data needs.

The spatial database in MAP for the PC is composed of one or more maps of a common scale describing a single study area. The database is currently set to hold a maximum of 99 maps. Each map is divided into grid cells; each grid cell contains a two-byte integer. Dimensions of the grid are defined by the user when the database is created, and are the same for all maps in the database.

Limitations imposed by disk capacity

The total data volume may be obtained by multiplying 2 (the number of bytes per cell) by the map dimensions, by the number of maps. According to this formula a single map layer 200 cells long and 250 cells wide would hold 50,000 cells and would be 100k in size. Taken to the 99-map limit, this single database would require 10MB of disk storage! Users must be aware of the tradeoffs between the number of cells in a map and the number of maps they expect to create, keeping in mind that the entire database must fit within the limits of available disk space.

Limitations imposed by memory size

The dimensions of the database affect program size because the program reserves internal storage space for two maps at any given time. Given the size of the remainder of the program, the size of the MetaWINDOW driver, and the 640k limit on available memory, the maximum number of grid cells in a map has been established at approximately 40,000 cells. An overlaid version of the program has been created as well, using the PLINK86 Plus[*] overlay linker. Because the overlaid version reduces the code in memory, a larger map size of approximately 64,000 cells can be accommodated.

Limitations imposed by screen capacity

The number of cells that can be displayed on the screen is limited by the screen dimensions of the graphics adapter. In the case of MAP for the PC, the current limit of 64,000 cells to a map does not exceed the screen dimensions of any of the display adapters supported by MetaWINDOW; overall screen capacity does not limit map size. However each of the map dimensions must be restricted if we wish to see the entire map at once. Map dimensions cannot exceed the corresponding screen dimensions without some loss of detail.

Impacts of map layer size on resolution and speed

The program scales the map to the screen dimensions. Therefore the map size determines the resolution of the displayed map. The more cells there are in a map, the more cells are squeezed into the screen area, and the finer the display resolution will be. The effect of various map sizes on screen resolution is illustrated in the GIS Lab Tech Report cited earlier.

Increasing the map layer size slows down the processing speed. This relationship is tested extensively in our system evaluation report. Using PFINISH[**], a performance analyzer program, we have clocked the speed of map data transfers between disk and memory, and the speed of selected GIS functions on several different map layer sizes.

CONCLUSIONS

The enhancements made to MAP for the PC have produced an easy to use program that can run under various hardware configurations. Extensive internal documentation has been added to make future modifications easier. The discussion of limitations imposed on map size by disk space, memory size and graphic devices gives a useful guideline for running the program effectively. The present version provides a stable teaching environment for students in GIS applications courses and a well documented tool for GIS design courses.

REFERENCES

Amundson, Sherry, 1987. Performance Evaluation of the Map Analysis Package: A Microcomputer-Based GIS. Technical Report 87-1, SUNY at Buffalo GIS Laboratory, Amherst, New York.

[*] PLINK86 Plus is a registered trademark of Phoenix Technologies, Ltd.

[**] PFINISH is a registered trademark of Phoenix Computer Products Corporation.

Metagraphics Software Corporation, 1986. MetaWINDOW Reference Manual. Scotts Valley, CA: Metagraphics Software Corporation.

Phoenix Technologies, Ltd, 1986. Pfinish User's Manual. Norwood, MA: Phoenix Computer Products Corporation.

Rasche, Brian and David J. Cowen, 1987. An Integrated PC-based GIS for Instruction and Research. Proceedings, Auto-Carto VIII. Falls Church, VA: American Society for Surveying and Mapping.

Tomlin, C. Dana, 1983. Digital Cartographic Modeling Techniques in Environmental Planning, unpublished Ph.D. dissertation, Yale University.

_________, 1986. The IBM Personal Computer Version of the Map Analysis Package. Report GSD/IBM No. 16, Graduate School of Design, Harvard University.

AN INTEGRATED PC BASED GIS FOR INSTRUCTION AND RESEARCH

Brian Rasche and David J. Cowen
Department of Geography and
Social and Behavioral Sciences Lab
University of South Carolina
Columbia, SC 29208

ABSTRACT

This paper describes the development of a totally PC based GIS system. The system utilizes the IBM enhanced graphics adaptor and a graphics kernel system. It expands on the original Map Analysis Package (MAP) by adding significant capabilities to input and display map layers. The system has been successfully utilized in several research projects and serves as the focal point for a graduate class in Geographical Information Systems.

INTRODUCTION

The Map Analysis Package (MAP) is probably the most widely used grid cell based geographical information system in existence (Tomlin Dissertation, 1983). For several years, the original mainframe FORTRAN code has been available for a nominal fee from the Yale University School of Forestry and Environmental Studies. Versions of the original algorithms now can be found in other Geographical Information Systems, such as The Map Overlay and Statistical System (MOSS). The simple English Language syntax and the powerful map algebraic approach, combine to create a versatile and comprehensive system for handling grid cell data bases. Furthermore, the system is quite well documented in the user manual and supplemental instructional materials, as well as in several research papers that describe its concepts and applications. The evolution of personal computers has provided an exciting framework for the creation of a self contained environment for MAP. The dual purpose of this paper is to describe a design strategy for such a PC based system, and to examine one resultant system that has emerged.

PC MAP ACTIVITIES

PMAP

One approach to the development of a PC version of MAP has been detailed in the work of Berry and Reed. These efforts have focused on the conversion of the original FORTRAN code into PASCAL and subsequently have resulted in the creation of PMAP. PMAP now includes several enhancements and modifications of the original version of MAP. Further, also is well documented and supports some interesting printer graphics. (Berry and Reed, 1986)

FORTRAN

Another approach to the development of a PC version of MAP has focused on the transfer of the original FORTRAN code into a desk top environment. Working at Harvard with the assistance of a grant from IBM, Tomlin completed the original work in this area. (Tomlin, 1986) The availability of this original source code, coupled with the emergence of powerful PC FORTRAN compilers and graphic kernel systems, have provided the impetus for the development of an enhanced PC version of MAP. This development has progressed through a cooperative effort at the University of South Carolina, the State University of New York at Buffalo (Marble, et al. 1987), Harvard and Penn State University.

DEVELOPMENT STRATEGY

The basic philosophy behind this cooperative PC development was to build upon the structure of MAP by improving the data input options, the user interface and the display capabilities (Fig. 1). The goal was to provide an integrated system that could handle any type of data input and greatly improve the options for displaying the output. The system was designed to be used on IBM-PC or compatible machines, with enhanced graphics adaptors (EGA), and support of a variety of graphic output devices through a graphics kernel system. Therefore, this hardware and software configuration should function in a widely available and affordable environment which supports acceptable graphic resolution and offers considerable output flexibility.

Operating Environment

The major enhancements to the original map operating environment consist of on-line assistance and a partitioned screen layout. The on-line assistance provides a list of extensive <EXPLAIN> options. Documentation for each of the 59 commands can be obtained through a simple <EXPLAIN> function that generates a screen of text describing the operation of the command and the proper syntax, including mandatory and optional modifiers (Fig. 2). The user interface divides the enhanced graphics adapter screen into three separate areas, a display window, a command line and a legend area. While this design enables the user to conduct all operations on a single monitor, a display window is left exclusively for graphics output. This environment has proven to be extremely friendly and responsive, thereby practically eliminating the need for external documentation.

Input Systems.

In order to be an integrated GIS, a system must be capable of handling any type of geographical data. Although MAP is strictly a grid cell based system, it is possible to convert any type of geographical features into a raster data structure. Points can be represented as individual cells, lines as contiguous groups of cells and areas as clusters of the same values. Continuous surfaces are matrices of numbers. The original version of MAP provided for the input of points <POINT>, linear features <TRACE> and gridded files <GRID>. While these options ultimately enable one to present any layer of geographical information, they required considerable preprocessing and manual encoding. MAP's major limitations related to the handling of geographical coordinate systems and the generation of gridded files directly from a polygon data structure.

Project. The vast majority of GIS data bases and existing maps can be linked to the Earth's surface by latitude and longitude, Universal Transverse Mercator (UTM) or State Plane Coordinates. A major enhancement to the MAP system incorporated the ability to convert easily between these systems. The FORTRAN subroutines necessary to accomplish this objective were extracted from the General Coordinate Transformation Program (GCTP) which are based on Snyder's Map Projections Used by the USGS. (Snyder, 1984) The command <PROJECT> reads a DOS ASCII file and outputs another file in a user specified measurement units. This routine eliminates much of the need to preprocess existing digital files before entering MAP, and allows the integration of a wide range of data bases. For example, data from digital line graphs, digital terrain models, Census polygons and digitized points all can be converted to UTM coordinate points and then input as separate layers.

PLPMAP. In order to benefit from the projection options, it is important to be able to read the resultant files and convert them into a common grid structure. The origin, cell size, and overall dimensions must be identical for each map and should be treated as user defined parameters. The existing MAP structure adequately handled points and gridded (matrix) data bases, however, it did not allow for true polygon input. The desire to accommodate all existing structures involved the inclusion of a general purpose point, line and polygon processor, <PLPMAP>. This routine reads any file output from <PROJECT> as either state plane or UTM coordinates and

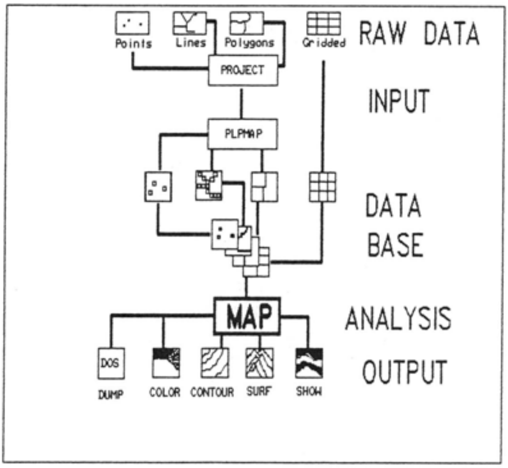
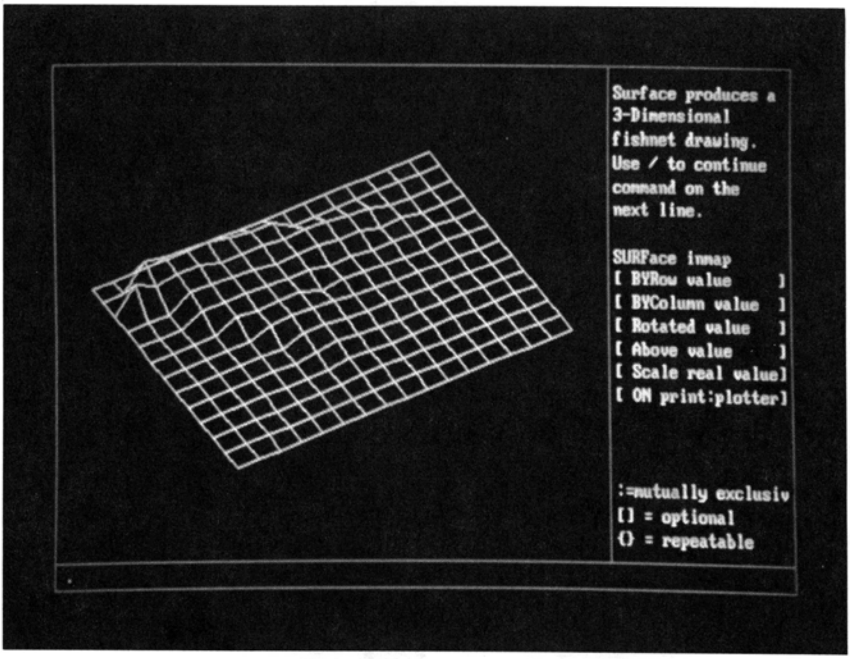


Figure 1. Flowchart of Integrated PC MAP System.



```
Surface produces a
3-Dimensional
fishnet drawing.
Use / to continue
command on the
next line.

SURFace inmap
[ BYRow value      ]
[ BYColumn value   ]
[ Rotated value    ]
[ Above value      ]
[ Scale real value]
[ ON print:plotter]


:=mutually exclusiv
[] = optional
{} = repeatable
```

Figure 2. SURFACE map and EXPLAIN command on EGA monitor.

subsequently converts them into either points, lines or filled polygons. The input files must be in SAS/GRAPH format (SAS Institute, 1981) and polygons must be closed loops, with embedded islands identified by a missing value code. The algorithm uses a centroid assignment procedure for converting polygons to specific grid cells. Presently, the SAS/GRAPH format is used by many organizations. Its straight forward structure can be developed from any topologically structured format. (Cowen et al, 1985) Gridded files, such as DTMS, can be handled with the existing <GRID> command.

In practice, the combination of <PROJECT> and <PLPMAP> have greatly expanded the input capabilities of MAP. In conjunction with a PC based arc node digitizing system, that includes an affine transformation, these procedures enable one to go easily, and directly from a digitizing tablet to a MAP dataset. Other special purpose formatting routines have been created to transfer any existing gridded data base, such as ERDAS or SURFACE II (Samson, 1975), into MAP.

Display Considerations

In order to create a more complete GIS, it was also necessary to improve the output display functions of MAP. In the original version, MAP was designed to generate character based line printer output using a <DISPLAY> command. The PC environment provides the additional capability of quickly generating color displays on a graphics monitor. Although Tomlin incorporated an interesting three dimensional perspective graphic display routine <SHOW> into the PC FORTRAN version (Fig. 3), there still remained the need to add other mapping options and direct the output to hardcopy devices.

COLOR. The major approach implemented for this aspect of the system involved the development of a general purpose <COLOR> procedure that incorporated a linkage to a graphics kernel system, GSS CGI, that supports several different display adaptors, and both raster and vector plotting devices (Graphics Software Systems, 1986). The syntax for the command is stated simply as: <COLOR> mapname on {PRINTER or PLOTTER}. The output will be routed to whichever device has been included in the configuration of the system. The <COLOR> procedure can display sixteen colors simultaneously on the enhanced graphics adaptor (Fig. 4). By incorporating a choice of three different palettes, the user may select a color scheme that is best suited to the particular data. For example, <PALETTE1> produces a variety of distinct hues, <PALETTE2> presents colors appropriate for continuous data, such as elevation; and <PALETTE3> generates color progressions of gray, blue, and red to yellow. The vector plotter option was designed to minimize pen changes while also optimizing the pen movements (Fig. 5). It is a surprisingly efficient output mode for raster data, with the plotting speed varying inversely with the homogeneity of the map.

The raster printer output mode of <COLOR> utilizes a series of patterns available with CGI. In order to utilize these patterns, a simple look-up table that assigns map values to pattern numbers was created. The first seven map values were assigned increasing gray scale density patterns, while the values eight through fifteen were assigned to various geometrical designs (Figs. 6 & 7). The <COLOR> command provides a versatile, attractive and easy to use procedure for generating an infinite variety of maps. Through the normal MAP overlay and renumbering steps, any number of maps can be combined and features assigned different colors and symbols. It cannot be over-emphasized that the final output map, whether presented on the monitor or as hardcopy from the plotter or printer, can portray an unlimited variety of information.

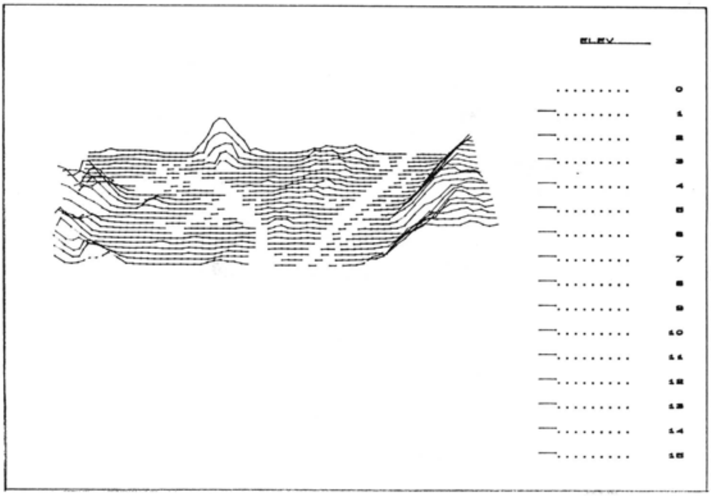


Figure 3. Example of output from SHOW command on vector plotter.

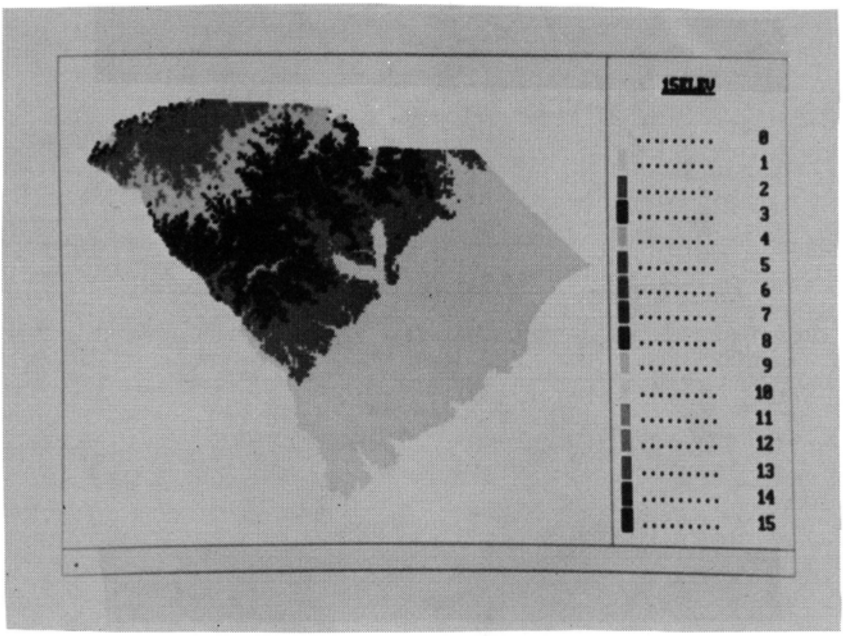


Figure 4. Elevation map of South Carolina displayed on EGA monitor with COLOR command.

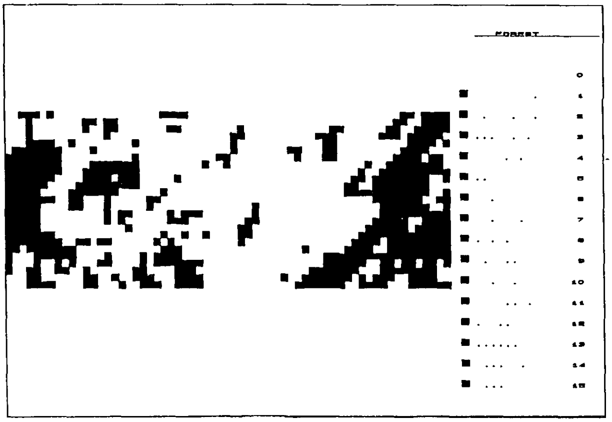


Figure 5. Single value vector plot created with COLOR command on plotter.

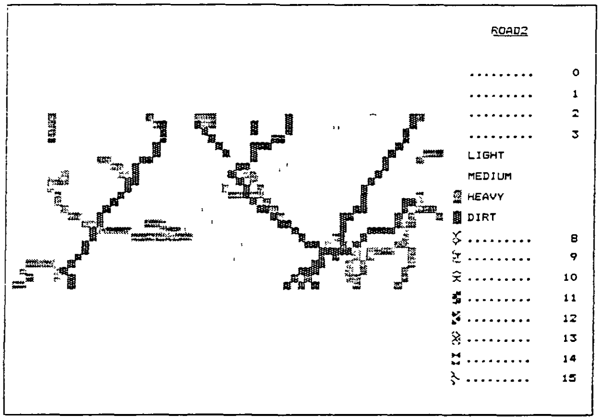


Figure 6. Dot matrix printer map generated by COLOR command. Note labels for values 4 through 8.

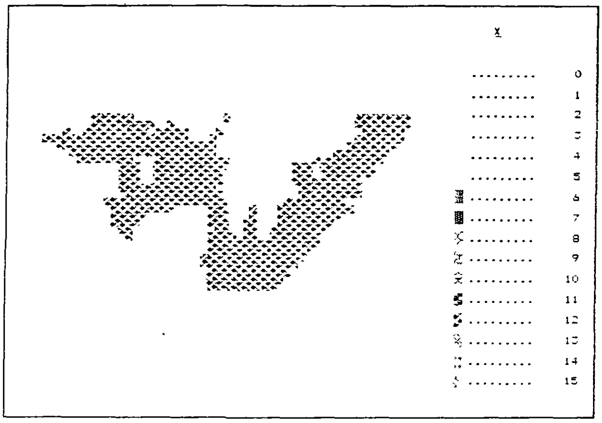


Figure 7. Single value COLOR printer map using only pattern.

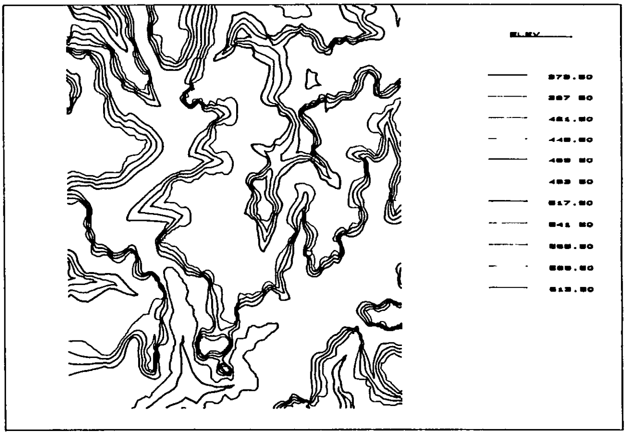


Figure 8. CONTOUR of elevation on vector plotter. Original data 218 x 180 for a 7.5 minute quadrangle.

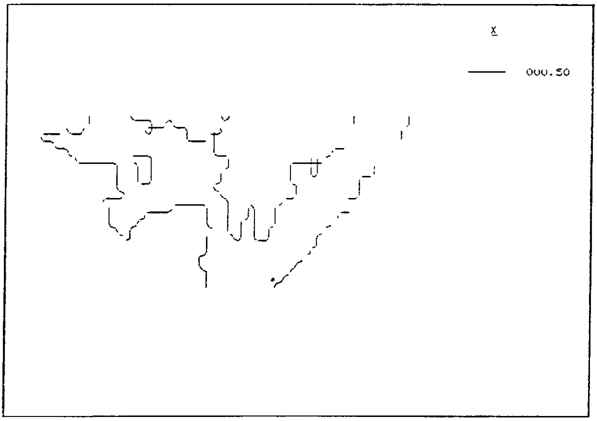


Figure 9. CONTOUR of single value on printer.

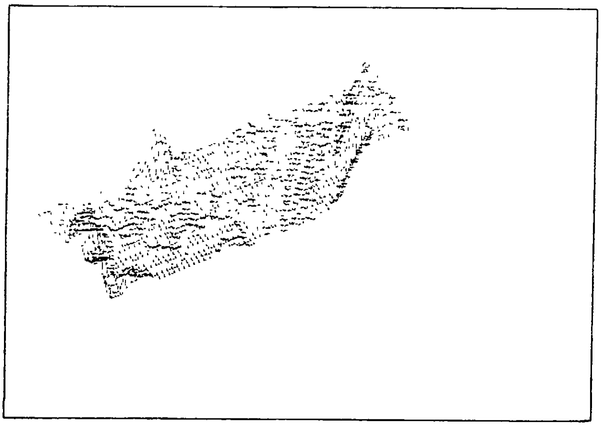


Figure 10. SURFACE map of elevation on printer.

The combination of a complete range of map inputs, unlimited overlay capabilities, available through ADDING, CROSSING, COVERING etc., and the RENUMBERING features of MAP, provides a truly universal approach to GIS manipulation and analysis. Furthermore, the limit of sixteen values on the EGA has not posed any serious constraints to the creative and aesthetic processes. The vector plot mode is limited to six pens. However, since the procedure calls the pens in sequential order, it is possible to pause and reload the pen carriage with six different pens. The user need only remember that any map is a matrix of numbers and that the <COLOR> command does not care how those numbers were created.

Contour. Another major enhancement to the display features of MAP was the inclusion of a <CONTOUR> option. <CONTOUR> threads contour lines between cells according to the intervals and ranges selected. As with the <COLOR> option, the display can be directed to the monitor, plotter or printer (Fig. 8). However, the routine was designed to function on raster output devices. While the screen and printer output are quite efficient, the vector plotter option is slowed down by frequent pen up and pen down movements, as well as pen changes. Further, the <CONTOUR> command provides a method of generating an outline map of original polygon or linear data bases (Fig. 9).

Surface. The third enhancement to the display features of MAP was the inclusion of a three dimensional display. <SURFACE> is a versatile, efficient routine that provides options for rotation, vertical scaling, altitude viewing and sampling of the matrix (Fig. 10). As with any three dimensional type of display, considerable experimentation with <SURFACE> is necessary in order to create a meaningful and aesthetically pleasing graphic.

Dump. The final addition to MAP consists of a utility for the export or transfer of individual maps from the data base to other systems. The <DUMP> command generates an ASCII DOS file in a row and column format that is handled easily by ERDAS and other image oriented systems, such as paint program. In practice this procedure has proven very useful for the creation of files that are "uploaded" to the mainframe version of MAP.

EVALUATION

The development of this system evolved over a nine month period. Since its completion, the system has been tested in several demonstration cases and several research efforts. One research project involved the analysis of various approaches for the extraction of stream locations and drainage basin boundaries from digital terrain data. The system provided a convenient and efficient environment for experimenting with different combinations of contours, aspect, slope, profile and stream operations (Fig. 8). Compared to the mainframe, the PC version of MAP saved the researcher hours of time. Furthermore, it produced much more usable and flexible output.

The other project involved the integration of several data layers in Pickens County, South Carolina. These layers consisted of Census County Divisions, Census tracts, USGS land use and land cover, a digital terrain model, water and sewer lines, water districts, industrial plants and major water discharge sites. The Census tracts and Census County Divisions were digital polygon files, and land use and land cover and DTM were gridded files, while all the other layers were digitized using a PC based system (Fig. 11). The entire data base was converted to 200 meter cells in UTM coordinates. Presently, this data base of 273 rows and 214 columns is being used county by planners to evaluate future industrial sites. They are learning how to use the system and are excited about their seminal experience with access to GIS capabilities.

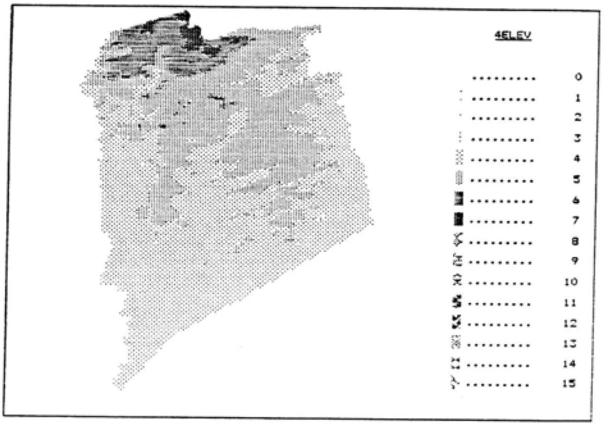


Figure 11. COLOR map on printer of 200 meter elevation data for Pickens County, S.C. Map is 273 rows by 214 columns.

FUTURE DEVELOPMENTS

Any GIS is an evolutionary system. By developing the graphics routines with a graphics kernel system, it will be possible to incorporate a variety of graphic devices, such as the laser printers and thermal plotters, as new drivers become available. It would also be feasible to move directly between MAP and a paint program for editing and color change. From a software viewpoint it would be desirable to functionally integrate the PC digitizing procedures into the same user interface.

REFERENCES

Berry, Joseph K. and Reed, Kenneth L. 1986, PMAP, The Professional Map Analysis Package Users Manual and Reference Guide, Spatial Information Systems, Omaha

Cowen, David J. et al. 1985, Alternative Approaches to Display of USGS Land Use/Land Cover Digital Data: Proceedings of AUTO CARTO VII, pp. 116-125

Earth Resources Data Analysis Systems, ERDAS, Atlanta, GA

Graphic Software Systems 1986, GSS*CGI Programmers Guide, Graphics Software Systems, Beverton, OR

Marble, D. F., Amundson, S., and Sindhu, J. 1987, Enhancement and Testing of a Microcomputer-based GIS for University Instruction: Proceedings of AUTO CARTO VIII, ACSM, Falls Church

Sampson, R. J. 1975, Surface II Graphics System, Kansas Geological Survey, Lawrence

Snyder, John P. 1984, Map Projection Used by the U.S. Geological Survey, USGS Bulletin 1532, GPO, Washington

Tomlin, C. D. 1983, Digital Cartographic Modeling Techniques in Environmental Planning, Ph.D. Dissertation, Yale University, New Haven

Tomlin, C. D. 1985, The IBM Personal Computer Version of the Map Analysis Package, Harvard Lab for Computer Graphics and Spatial Analysis, Cambridge

IDRISI : A COLLECTIVE GEOGRAPHIC ANALYSIS SYSTEM PROJECT

J. Ronald Eastman
Stacy M. Warren
Graduate School of Geography
Clark University
Worcester, MA 01610.

ABSTRACT

IDRISI is a grid-based geographic analysis system, developed at Clark University, that is designed to provide the focus for a collective program of system development and exchange. The core consists of a set of independent program modules that act upon a simple and easily accessible data structure. Individual researchers can thus add new modules in any computer language so long as this simple data structure is maintained. In order to provide the greatest possible flexibility, both real and integer data types are supported, along with ASCII, binary, and run-length encoded data files. Similarly, continuous images such as remotely sensed data and digital elevation models are handled as readily as categorical map coverages. The core program set, written in PASCAL, provides a series of fundamental utilities for the entry, storage, management, display and analysis of raster images. In addition, a ring program set has been established to provide a group of analytical tools commonly associated with grid-based geographic information systems. Currently available in inexpensive development versions for CP/M, MS-DOS and VAX-VMS systems, it is hoped in the near future to establish an on-line bulletin board through which users can exchange experiences and new program modules.

INTRODUCTION

Over the past two decades geographic information systems technology has developed rapidly. However, access to that technology has not been as forthcoming. Software costs have tended to be high, hardware needs specialized, and individual program development difficult. In addition, problems requiring small-scale analysis are often overwhelmed by large-scale systems, and research-oriented projects commonly require analytical capabilities well beyond those of a "standard" configuration (eg. Olsson, 1985). In an attempt to address these problems, a geographic analysis system named IDRISI has been developed at Clark University over the past year. Presently available in development versions for VAX-VMS, MS/PC-DOS, and CP/M-80/86 operating systems, the system supports a wide range of readily available microcomputer systems.

DEVELOPMENT PHILOSOPHY

IDRISI can perhaps best be characterized as a research analysis system rather than a large-scale inventory tool. Consisting of a set of completely independent program modules, IDRISI was never designed to become a single comprehensive program. Rather, it was intended to establish an environment for the creation and development of individual project-related modules by independent researchers on the basis of its simple and easily accessible data structure and core program set. By establishing a network whereby users can share experiences and program development, it is hoped that the system can then grow in organic response to the needs of the user community.

Given these general objectives, the development of IDRISI has been based on four primary principles :

1. the system should be available at low cost, using readily accessible technology;
2. data integrity should be maintained at all times;
3. data accessibility and system flexibility should have precedence in program design; and
4. the system should naturally accommodate the display and analysis of continuous images as readily as categorical map types.

The effects of these principles can be seen in all aspects of the system, ranging from data structure, to program logic, to image display.

DATA STRUCTURE

Perhaps the most fundamental consideration in the design of the system was data structure. IDRISI is a grid-based geographic analysis system, and thus processes information about image cells in a map-like matrix. However, unlike some systems which require that these data be stored as two-byte integers, IDRISI accommodates both integer and real data types. Furthermore, there is no limit to the size of the grid employed, and grid dimensions can be changed at any time by either windowing or concatenating new sections. Each coverage is stored as a separate file, with a single-field record for each cell. Both ASCII and binary storage structures are supported for these files, along with the ability to transform from one to the other when needed. In addition, run-length data compaction is supported as a storage structure for categorical (integer) coverages.

Given that the data files are not structured according to the particular characteristics of any project, each data file has an accompanying documentation file (automatically created) that records information about the data type as well as other features such as the number of rows and columns, title, legend (if any), ground resolution, and the like. In all operations, IDRISI programs use this information, along with normal mixed-arithmetic rules, for determining the data type and geographic structure of the result.

The primary motivating factors behind the choice of data structure were data integrity and data accessibility. There are many geographical operations that require the precision and range of real numbers. Given that the system was envisioned as a small-scale analysis system rather than a large-scale inventory tool, the extra storage requirements were felt to be negligible compared to the enhanced flexibility and essential integrity that real numbers would afford.

In order to give maximum flexibility to the import, export and modification of data, it was felt imperative that the user have direct access to data files at all times. It was further considered that many users who might wish to create their own modules would be most comfortable programming with ASCII format files. It is for these reasons that the default data format supported is ASCII. While not the most efficient format, it combines the desirable qualities of simplicity and accessibility.

The ASCII format works quite well when the number of digits per cell are few (such as with LANDSAT imagery and most categorical maps). However, when real numbers are extensively used, serious storage problems can result. Version 1 of IDRISI simply allowed the conversion of ASCII to binary as a storage compaction technique. However, Version 2 (in progress as of Nov. 1986) incorporates binary as a fully supported data format. As with the data type, data format is recorded in the accompanying documentation file, and all operations in Version 2 preserve the prevailing data format. Thus, users can work consistently in one format or the other, without ever having to switch. However, the CONVERT facility will always allow a change from one to the other.

PROGRAM STRUCTURE

In keeping with the notion that IDRISI should be a program that grows organically in response to user needs, the decision was made that it should consist of a series of independent program modules rather than a single unified program. The effect of this is that new programs can be added that are entirely independent of other programs in the IDRISI set. Furthermore, the language in which these programs are written is irrelevant to system operation. New programs need only adhere to the simple structure of the data files. Furthermore, for programming simplicity, documentation files can be ignored. A module exists to create these files as a separate operation when needed. Individual users can thus add new capabilities in the programming language of their choice, without reference to other module operations or the housekeeping functions normally required for efficient operation.

In order to gain the widest possible circulation among currently available microcomputers, it was decided to create the core program modules in Turbo PASCAL. For larger-scale applications, a VAX PASCAL version has been written concurrently with the micro versions. However, some differences exist between these language implementations. Therefore, to facilitate this development, a series of generic routines (for the most part input/output) were written to give some commonality between the two versions of PASCAL. Each module thus has access to an internal subroutine library which executes version-specific commands. The program code in the modules themselves appears identical from version to version.

The experience with Turbo PASCAL has been excellent. The interactive nature of the editor and rapid compilation have greatly speeded program development. The resulting code has also been found to be compact and efficient. Although there are some shortcomings with respect to the size of data and code segments it can handle, this has not proved to be an issue in this project. We have therefore adopted Turbo PASCAL as the primary language for system development.

PROGRAM CAPABILITIES

IDRISI program modules fall into one of three groups:

1. core modules, providing fundamental utilities for the entry, storage, management, display and analysis of raster images;
2. ring modules, providing extensions commonly associated with a major analytical mode (such as GIS or Image Processing); and
3. peripheral modules associated with specific research projects.

IDRISI's core program modules (Table 1) provide the primary foundation upon which individual research modules can be built. As can be seen in Table 1, there is a strong emphasis upon data management, retrieval and display in this set, with the original design objectives playing a strong role in determining the particular mix of capabilities offered. For example, the data accessibility and system flexibility objective can be seen as a major influence in the design of the data management modules, while the desire to be able to analyze and display continuous images played a strong role in configuring the retrieval and display modules. Additionally, the core set includes some basic facilities for analysis that were considered to be fundamental to all related applications. These are the RECLASS, SCALAR and OVERLAY operations.

Initially, it was intended only to develop those modules envisioned for the core set. However, the decision to apply IDRISI to an ongoing GIS project for the Clark University Program for International Development, quickly precipitated the formulation and writing of a GIS ring (Table 2). The GIS ring includes analytical operations which are, for the most part, typical of grid-based geographic information systems. The capabilities chosen for the core and GIS ring, and their particular characteristics, evolved from a careful consideration of both theoretical classifications of GIS capabilities (eg. Berry, 1984; Dangermond, 1983; Dangermond and Freedman, 1984; Olsson, 1985; Tomlinson, 1976), and practical experience with the ODYSSEY (Morehouse and Broekhuysen, 1982), UDMS (HABITAT, 1985), and Map Analysis Package (Tomlin, 1980) systems. The Map Analysis Package was particularly influential in establishing this set, as can be seen in the group of image-wise operations such as DISTANCE, PATH, VIEWSHED and WATRSHED. Olsson's (1985) innovative study of desertification in the Sahel was also a strong influence, providing the impetus for including the INTERPOL and HINTRLND modules.

Although the IDRISI system was primarily envisioned as a geographic information system, the need to include facilities for treating continuous data, along with a desire to provide fundamental capabilities for incorporating remotely sensed data, naturally led to a consideration of image processing operations. Although the Image Processing ring has not yet been comprehensively addressed, a number of modules have been created in conjunction with projects using remotely sensed images (Table 3). In addition, it should be noted that the core program set allows a wide range of image processing operations to be undertaken. Indeed, on the understanding that a common conceptual and procedural basis links all raster operations, all IDRISI operations are considered to act on images.

Perhaps the most basic facility that reflects this character is the IMAGE program which accommodates the display of continuous images. In order to provide an inexpensive system with reasonably high resolution, a half-tone technique was created using groups of adjacent pins on readily-available dot-matrix printers (Figure 1). The drivers vary from one printer to the next, but typically involve a 6 by 3 group of pins (overlapping by 50% in the x direction) to produce a square pixel of about a millimeter in width. An 8.5 by 11 inch page can thus usually accommodate a 256 by 192 image window. The 6 by 3 pin configuration easily accommodates 32 grey levels -- a resolution quite adequate for many purposes. In order to prepare images for display, the program module HISTO produces image histograms while STRETCH can be used to undertake a linear contrast stretch and output a new image with a range from 0 to 31. For very small image windows, the program module EXPAND can be used to enlarge the display window.

Among the core analytical operations, OVERLAY also reflects this image processing character. The OVERLAY module was designed to encompass all operations that produce a new image from more than one original. These include the ability to add, subtract, multiply, divide, minimize, maximize, cover and exponentiate on a pixel-wise basis, with boolean operations being achieved with binary images. Since the divide operation accommodates the ratio operation commonly used in many natural resources applications of remotely sensed images, a "normalized ratio" was also included. Here, the difference between two images is divided by their sum -- an operation commonly used in the derivation of vegetation indices using the red and near-infrared bands of multispectral imagery.

The FILTER operation was originally included as a neighborhood characterization process, similar to the SCAN operator in the Map Analysis Package GIS. However, given this evolving image processing character, the facility was extended to allow the convolution of any user-defined 3 x 3 filter over the input image. In addition, options include pre-defined mean, median, mode, edge-enhancement, low-pass and high-pass filters.

As can be seen from these examples, the IDRISI system is intended as a general purpose raster processing system. As mentioned, no work has been specifically directed towards the development of an Image Processing ring. However, rudamentary image analysis can be achieved with the core set. For example, a parallelepiped supervised classification can be achieved by using the STRETCH, IMAGE and WINDOW operations to first view the image and extract training areas. The HISTO module can then be used to examine spectral response patterns (the SCATTER module has since been added to aid this process), with the RECLASS module being used to extract the class range within each band as a binary image. The multiply option of the OVERLAY module can then be used to combine the results from each band, and thereby derive the final classification.

PROCESSING LOGIC

Given the fact that the system is designed to run on a variety of microcomputers of varying capability, all IDRISI modules use as little random-access memory as possible. This is not simply a concession to current microcomputer technology (which is changing rapidly). It is implicit in the design philosophy that the size of image that can be handled should be constrained as little as possible by memory availability. Therefore it was considered that memory is better employed on processing-buffers than image storage.

IDRISI operations fall into one of three categories: pixel-wise, region-wise or image-wise. Pixel-wise operations, such as those involved with almost all of the core modules, require storage for only one data value at a time. Region-wise operations such as the convolution operator in FILTER, or the neighborhood functions in SURFACE and AUTOCORR temporarily store a moving group of scanlines at a time. Memory requirements are thus kept to a small multiple of the number of image columns. At present, the microcomputer versions of these modules are restricted to images of 2000 columns or less, but an unlimited number of rows. This could be increased considerably with a large-memory model version of PASCAL (anticipated with the next release of Turbo PASCAL). However, images of such a magnitude would clearly approach the manageable limits of most current microcomputer systems. For wider images, either the image can be TRANSPOSed or the WINDOW module can be used to extract sub-images for processing.

A more challenging group of operations are those that require random access to the entire image. These include the image-wise operations of DISTANCE, PATH, VIEWSHED, WATRSHED and HINTRLND. An obvious alternative to core-resident random-access memory is the use of either floppy or hard disk-based random access files. Since disk space is a relatively cheap commodity, this is clearly a viable alternative for microcomputer-based systems. However, the mechanical operations required by this process can significantly reduce processing speed. Algorithms therefore have to be chosen with care. The DISTANCE module, for example, which calculates the proximity of each pixel to the nearest of a set of target cells, uses an "advancing-front" technique to alleviate this problem. In two passes, one through the sequential data file, and a second backwards through a random-access intermediate file created during the initial pass, the program computes the distance from every cell to the nearest non-zero cell in the target image. The process is very much like moving a directional processing "squeegee" down, and then back up the image. The random-access disk file is thus used only once --to move back up the image.

As a result of these considerations, IDRISI is clearly disk-intensive. However, by increasing the size of the buffers associated with text (ASCII) files, and using high-speed block transfers to processing arrays for binary files, efficient operation can still be achieved.

PRESENT STATUS and FUTURE DEVELOPMENT

At present, IDRISI is available in development versions for most CP/M, MS/PC-DOS, and VAX-VMS systems. It is intended that the pricing of the system should always reflect no more than the costs of maintenance and distribution. Furthermore, it is our hope that the system will provide a focus for independent and collective program development. To this end, we plan to establish an on-line bulletin-board system through which users can exchange software and experiences. Through this medium, users will be able to access news about the system, immediate software updates, and advice on system use from both fellow users and system developers. Those that develop their own modules can upload them in order that they may be shared by the user community. User uploads will then be reviewed and, if necessary, modified for inclusion in the standard IDRISI set.

REFERENCES

Berry, J.K., 1984, Cartographic Modelling: Computer-Assisted Analysis of Maps, in Map Analysis Package Academic Materials, Yale School of Forestry and Environmental Studies, New Haven, CT.

Dangermond, J., 1983, A Classification of Software Components Commonly Used in Geographic Information Systems, in Design and Implementation of Computer-Based Geographic Information Systems, D.J. Peuquet, J. O'Callaghan, ed., IGU Commission on Geographical Data Sensing and Processing, Amherst, NY.

Dangermond, J., Freedman, C., 1984, Appendix 1 of A Conceptual Model of a Municipal Data Base, in Basic Readings in Geographic Information Systems, D.F. Marble, H.W. Calkins, D.J. Peuquet, ed., SPAD Systems Limited: Williamsville, NY, p. 2-91 - 2-114.

HABITAT (United Nations Centre for Human Settlements), 1985, Data Management: Urban Data Management Software (UDMS) User's Manual, Second Edition, United Nations, HABITAT, Nairobi, Kenya.

Morehouse, S., Broekhuysen, M., 1982, ODYSSEY User's Reference Manual, Harvard Univ. Lab. for Computer Graphics and Spatial Analysis, Cambridge, MA.

Olsson, L., 1985, An Integrated Study of Desertification, Lund Studies in Geography, Ser.C, No.13.

Tomlin, C.D., 1980, The Map Analysis Package, Yale School of Forestry and Environmental Studies, New Haven, CT.

Tomlinson, R.F., Calkins, H.W., Marble, D.F., 1976, Computer Handling of Geographical Data, The UNESCO Press, Paris, France.

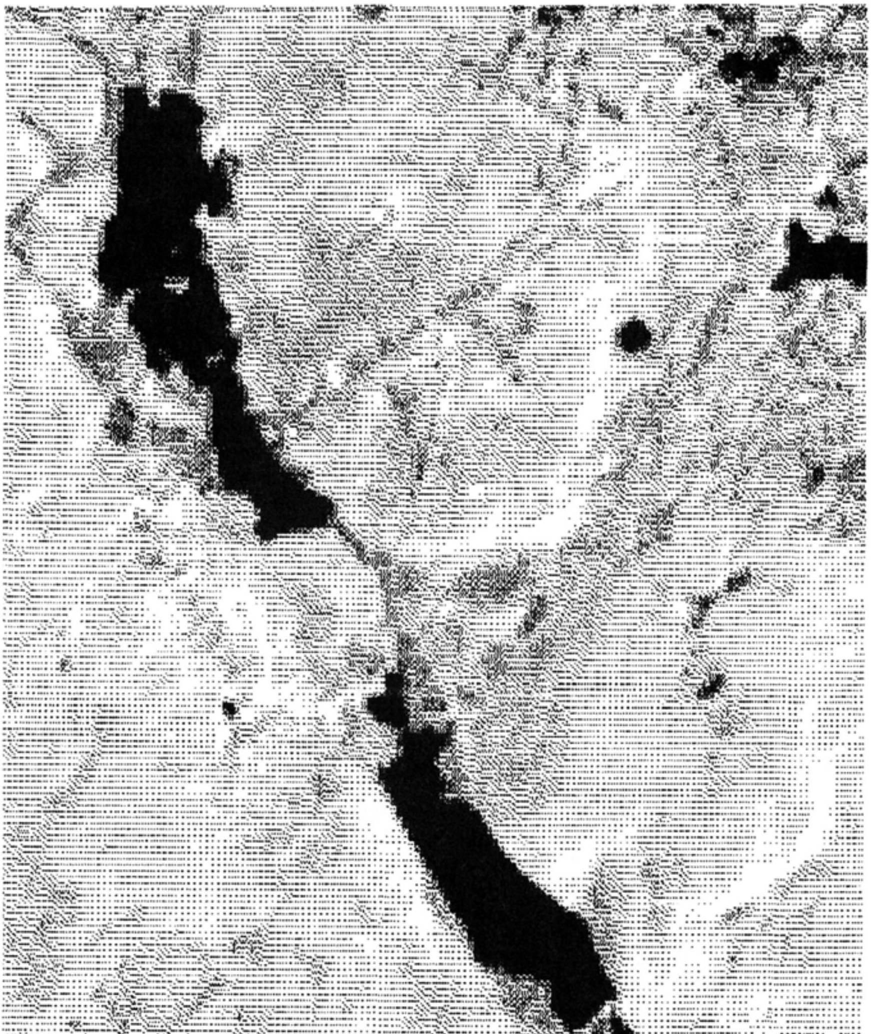


Figure 1 : An example of output from the IMAGE program module. The scene is a portion of a Thematic Mapper Band 4 (infrared) image, near Worcester Massachusetts.

TABLE 1 : IDRISI PROGRAM MODULES IN THE CORE SET

A. Data Entry Modules

INITIAL	Initializes a new image with a constant value.
UPDATE	Keyboard entry / update of image data.
POINTRAS	Point-to-Raster conversion.
LINERAS	Line-to-Raster conversion.
POLYRAS	Polygon-to-Raster conversion
INTERPOL	Interpolates a surface from point data using either a weighted distance or potential surface model.

B. Data Storage Modules

CONVERT	Converts data files from ASCII to binary or vice versa.
PACK	Converts integer data to run-length encoded storage format.
UNPACK	Converts run-length encoded data to sequential format

C. Data Management Modules

DOCUMENT	Creates a documentation file for a new imported image file, or revises the document file of an existing image file.
CONCAT	Concatenates two images to produce a larger image.
TRANSPOS	Transposes the rows and columns of an image.
WINDOW	Extracts a rectangular subimage.
QUERY	Extracts pixels designated by an independent mask into a sequential file for subsequent statistical analysis.
EXPAND	Enlargens an image by pixel duplication.
CONTRACT	Reduces an image by pixel aggregation or thinning.

D. Data Retrieval and Display Modules

IMAGE	Produces a grey-scale image (up to 32 levels) using a half-tone procedure on dot-matrix printers.
DISPLAY	A "universal" display routine using ASCII characters. Large images can be displayed as a series of sub-images. Legends are produced using the image documentation file, and area statistics are given for each category.
VIEW	Allows direct examination of any portion of an image. Output precision is user-specified.
HISTO	Produces histograms of image file values. In addition to the graphic output, a numeric output includes proportional and cumulative frequencies along with simple statistics.
STRETCH	Produces a linear contrast stretch in preparation for image display using the IMAGE module.

E. Analytical Modules

OVERLAY Undertakes pixel-wise addition, subtraction, multiplication, division, and exponentiation of paired images. Maximum, minimum, "normalized ratio" (eg. vegetation index), and "cover" (logical OR) are also supported. Other boolean operations such as logical AND, XOR, EQV, NOT and IMP are supported through various binary image overlay combinations.

SCALAR Adds, subtracts, multiplies, divides, and expotentiates pixels by a constant value.

RECLASS Reclassifies pixels by equal intervals or user-defined schemes.

TABLE 2 : IDRISI PROGRAM MODULES IN THE GIS RING SET

SURFACE Produces slope gradient and aspect images from a surface.

AREA Creates a new image by giving each output pixel the value of the area of the class to which the input pixel belonged.

PERIMETR Creates a new image by giving each output pixel the value of the perimeter of the class to which the input pixel belonged.

GROUP Classifies pixels according to contiguous groups.

DISTANCE Calculates the distance (proximity) of each pixel to the nearest of a set of target pixels. Distances can be weighted by an auxillary friction (eg. cost) surface.

PATH Finds the shortest path between two points, with the origin specified on a target image, and the destination specified as the lowest point on a surface image (eg. cost).

VIEWSHED Creates an image of all points visible from a target over a given surface.

WATRSHED Creates an image of all points uphill of a target.

HINTRLND Determines the supply area dominated by point demand centers.

AUTOCORR Computes Moran's "I" autocorrelation statistic for an image.

TABLE 3 : IDRISI PROGRAM MODULES IN THE IMAGE PROCESSING RING SET *

* This set has not yet been comprehensively planned. Modules listed only include those existing as of November, 1986.

FILTER Convolves an image with a digital filter. Mean, median, mode, edge-enhancement, low-pass, high-pass and user-defined filters are accommodated.

SCATTER Produces a two-band scatter plot.

RADIANCE Converts LANDSAT MSS and Thematic Mapper DN values to spectral radiances.

CLASSLESS CHOROPLETH MAPPING WITH MILLIONS OF COLORS: A DEVELOPMENTAL SYSTEM

James R. Carter, Ph.D., Carolyn S. Bolton, and M. Elizabeth Wood
Geography Department, University of Tennessee
Knoxville, TN 37996-1420

ABSTRACT

Using a Vectrix color display system driven from an IBM PC/XT a classless choropleth mapping package has been developed. The system has evolved and there is no master plan to build a finished system. The display system has a resolution of 672 x 480 and can display 512 simultaneous colors from a palette of 16.7 million colors. The choropleth mapping package that has been developed on this system has been the product of many experiments by the author and his students. The paper traces the evolution of the package and discusses how the system is being employed in the teaching of computer graphics, teaching of cartographic design, and the conceptionalization of many expert systems to sort through many of the options available.

INTRODUCTION

When Tobler (1973) stated that computer technology would give us the ability to display a continuum of shades and that we would not need to deal with class-interval questions anymore, he was most prophetic. Using a medium resolution display capable of addressing 8 bits per primary color and displaying 9 bits per pixel provides the ability to generate more shades than the eye can differentiate and is a good example of the technology that Tobler anticipated. With such a system 256 gradations of shades can be displayed between any of the pure primaries or combinations of the pure primaries--black, white, red, blue, green, magenta, yellow, and cyan. By adding or subtracting tints of any primary to the basic gradation, an almost unlimited palette of gradations can be selected to portray any given choropleth distribution.

Three years ago the Geography Department at the University of Tennessee, Knoxville purchased an IBM PC/XT and a Vectrix Corporation VX-384 color display system. The VX-384 is a separate frame buffer residing external to the PC and driven by the PC through a serial port. The VX-384 outputs RGB signals to an Electrohome monitor. Recently, a duplicate version of this system was purchased using a Vectrix VXPC board set driving a NEC Multi-Sync monitor. Two years ago the Vectrix PaintPad system was added to the VX-384 system using a Summagraphics digitizing tablet. The newer system includes PaintPad under the control of a mouse.

All of the software that has been written was developed on the VX-384. The VXPC was configured to make use of the same software. All of the programs accessing the Vectrix VX-384 display are written in Interpreted BASIC, because the interpreter was readily available, the author and students had at least a rudimentary working knowledge of the language, and it was very easy to make small changes and quickly see the effects of the change. No thought has been given to changing to another language because the response times of the VX-384 are slower than the rate at which BASIC produces commands going the the 384. In some cases we have had to build delays into the software because the output from the interpreter overwhelms the frame buffer. By staying with the interpreted BASIC new students have been able to learn to program by building on the existing programs. However, as we consider extending the system to incorporate some expert sytems we know we will have to work in another language.

MATRICES OF COLOR POSSIBILITIES

To gain an understanding of how to address the full range of 16.7 million colors in the lookup table, a program was written to display simultaneously 8 matrices consisting of 8 rows of color squares by 8 columns of color squares. This, of course, pushed the display capabilities to a maximum since 8 x 8 x 8 = 512. Assigning the RGB primaries to range from a minimum of 0 to the maximum of 255 in eight equal steps gives a good overview of all color gradations that are possible with the system. By assigning a more limited range on any or all of the primaries, it is possible to 'zoom in' on a part of this three-dimensional color spectrum. The use of CTABLE has given us some insights into working in color space. These matrices show that there is a perceptual effect of one shade appearing to be lighter where it abuts a darker neighbor and appearing to be darker where it abuts a lighter neighbor. Another insight is that it is hard to find a good variety of gradations of browns and oranges on a RGB system even thought there are millions of colors to pick from.

POLYGON BOUNDARIES AND CENTROIDS

Coordinate boundary files have been employed for the counties of Tennessee and the states of the U.S. Both of these files were taken from files already used by the authors. All of these files were in a polygon format and were kept in this format, even though they require redundant writes for all interior boundaries. The outlines of the areas to be mapped were scaled to the screen and the coordinates were transformed using a simple scaling and translation routine. The senior author selected the scaling and position of the map areas and no provision exists in the package to permit the user to scale, translate, or rotate the subject on the screen. It is felt that there are more than enough variables for the user to address in the present system without having to deal with scaling and figure/ground relationships.

Many of the boundary files we started with were excessively detailed for the resolution of the screen and the scale of the map. Two students wrote an interactive program to reduce the number of points in a polygon boundary. This program writes a single polygon boundary on the screen and then steps around the boundary going point to point in response to touchs of the keyboard. At each point the user has the option of eliminating that point or retaining it. The process is subjective but it permits the user to build a visually appealing set of boundaries. A subjective line generalization procedure is not appropriate for boundaries stored in a polygon format because the subjective generalization of one section of an interior boundary is not always the same on both representations of that boundary. This program would work much better if the boundary files were in a chain format. But, for this situation, the program produced generalized files after a few iterations. All slivers were removed in these iterations.

The VX-384 employs a polygon fill routine that starts from any seed within the polygon. The polygon boundaries are written to the screen in a color different from the background. For each seed a screen coordinate has to be specified that falls within the polygon. The process of selecting seeds within the polygon was done manually for one cartographic data set and was done by calculating geographic mid-points for another cartographic data set. Where the centroids were calculated, corrections had to be made for a few irregularly shaped polygons. For Michigan, with its Upper and Lower Peninsulas, two seeds were required. Maryland required 3 seeds because in its western extent the width of the state is so narrow that in two places the north and south boundaries occupy neighboring pixels which has the effect of creating separate polygons relative to the fill primitive. In the mapping program the file of centroids is read each time new data are assigned to the polygons.

A number of data sets have been put together to use in this package. All data sets have been organized to have a one line entry for each state or county with the FIPS code and an alpha name in addition to one or more variables. In each case a separate input routine has been written to read each data set. This is quite cumbersome and we are working to build a single routine that will read from any of the data sets in an efficient fashion.

SELECTING COLOR SEQUENCES

The values are read into an array and in the process the minimum and maximum values within any data set are identified. These two extremes are used to calculate the range of data. In a FOR: NEXT loop the individual values for each state or county are assigned a position along the linear continuum between the extremes, a shade is calculated proportional to the position along the continuum, the coordinates of the seed are read, the cursor is moved to the seed position, and the polygon is filled with the appropriate shade. The program was identified as producing classless maps but in reality the data are divided into classes--128 to be exact. Where the shades are juxtaposed in a non-graded sequence such as occurs on the choropleth map no one can distinquish any single shade across the map. Thus, in reality, 128 shades form a continuum. In fact, 64 shades, or perhaps even 32 shades, would be no less effective when used in a choropleth map program of this type. The graphics system permits 256 separate shades to be defined, but only 128 were used in the program originally and there has been no cause to use the maximum number.

For a long time we used the default color look-up table as we wrote the first map to the screen, even though it produced meaningless maps. Most beginners had great trouble understanding what the system was all about when they began with this random display of many colors. So, we set up the color look-up table so that the first map seen on the screen employs a continuum from black to white. Beginners find it much easier to understand the function of the system when they are presented with this preselected example of color choices. And, in our opinion, the gray-scale map should be seen by all users anyway for it is a great place to start because it is neutral in terms of colors. In many situations, these gray-scale maps are more powerful than many of the colored maps.

Kimerling (1986 and 1985) has found that viewers do not differentiate the dark end of a linear gray shade continuum ranging from black to white on a CRT any better than they do on the printed page. This suggests that we should use a non-linear continuum with larger steps at the dark end of the scale and smaller steps in the middle and at the light end of the scale. However, once the first display is put on the screen, the user is given control of a program called CTABLE, which permits the user to set the colors at the extremes of the continuum. The user may choose to go from any color combination to any other, i.e., a light yellow-green to a golden brown. With such a color combination there would be no true dark end of the continuum, Thus, at the moment we see no reason to employ anything other that the linear division of the gray scale.

CTABLE is a color look-up table that walks the user through the process of selecting color schemes for the shading of the polygons, for the background areas, and for the boundaries and titles. We have found that after about 15 to 30 minutes with the program a person develops a feel for color space in an RGB system and starts to seek combinations that interest them. CTABLE starts by asking the user to specify Low Red, Low Green, and Low Blue as numbers between 0 and 255. Next the user is asked to specify High Red, Green, and Blue as numbers between 0 and 255. Within 30 seconds the map on the CRT changes to the new color sequence. The user can then change the color sequence again or can change the background color or can change the boundary color. Background and boundary colors are specified by giving a value between 0 and 255 for each primary. Some of the possible combinations are given below:

Numeric values of the primaries Red, Green, and Blue give

R	G	B		R	G	B	
0	0	0	to	255	255	255	gives black to white
255	0	0	to	255	255	255	gives red to white
255	0	0	to	0	0	255	gives red to blue
255	80	40	to	40	80	255	gives bright red to bright blue
255	100	100	to	100	255	255	gives salmon to bright blue
0	50	200	to	255	200	100	gives blue to tan

A variation of CTABLE asks the user to specify a range of 0 to 100 so that the user selects the intensity of each primary as a percentage. We have not noticed any improvement in people's ability to understand the task when working with the 0-100 scale, so we have tended to stay with the 0-255 scale. Quite a few users have been able to understand the bit configuration that is in operation when the 0-255 scale is used.

Providing a legend for the continuous tone maps has given some difficulty. Ideally, we think the legend should be a bar showing the full gradation used in the map. So far, we have positioned the bar in what looks like a balanced position relative to the map. We have made no provision for the user to shape his or her own bar legend or to place the bar where they feel it is appropriate. This may be an area for the next inventive student to take on. Assigning numbers to the bar so that the user can interpret values from the map is another problem. The easiest way to do this is to add the numbers using the PaintPad program, but this is out of keeping with the nature of an automated system. The numbers used should mark the end points of the bar in fairly round numbers so that the user can relate to the data, i.e., 16 - 48 rather than 16.17 - 47.69. Then there should be one or two additional numbers placed along the bar at significant points. What those significant numbers might be will vary with the range of values. Another problem of the same type is that of selecting, sizing, and placing a title. So far, we have employed the PaintPad program to do this and it works fairly well. But again, this is out of keeping with a fully automated system.

ADDING INTELLIGENCE TO THE SYSTEM

There are a number of places where expertise could be incorporated into such as system, particularly if the system is to be used as a decision support system. We have been looking at such situations and they have become a focus for our study of expert systems (Robinson and Jackson, 1985, and Pfefferkorn, et. al., 1985). The division of the legend bar is one such place where expertise could be incorporated into the process. The selection and placement of titles is another. Still another might be in selecting palettes that might be tuned to the particular user. A persons using such a system to get map information is probably not interested in playing with the 16 million colors. What this person really wants is a map that communicates that message to tell him or her what they think they want to know. What colors will be best to do this? We may never be able to find out which colors are best but we think we could find out that certain color sequences are more effective for a particular individual than are other sequences. If this is true, then a system could be tuned to the decision maker using the system. We think it might be possible to test the user of a decision support mapping system in the early part of a session so that subsequent activities will take place employing the more effective color palettes. Building such an expert system is beyond our immediate goals, but it is educational to contemplate how such interactivity might be incorporated into the map reading process.

Data selection is another area where expertise needs to be brought into the thematic mapping process, particularly if the mapping is being done as part of a decision support system. As a point for discussion, assume we have a data base with 100, 200, or 500 variables that might be called up by title or index number. How does the user sort through this collection and select the one or two variables that might address the topic he or she wants to address? Many times the senior author has worked with persons who had specific ideas of what they wanted to see in the way of a map. After showing these persons a map or two and engaging them in some directed conversation, it often comes out that there is another map that is more in keeping with what they really want. This same problem will occur as the general public has access to some of the data bases that supposedly are now or soon will be a reality. Is there any way to transfer that knowledge-based dialog to an interactive program? (Smith, 1986). To help us understand this basic concern, we have created a simple data set that has enough power to let us explore the process.

For the 95 counties of Tennessee we assembled three variables: 1980 county population, 1985 county population, and the land area of each county in square miles. Such variables are likely to be found in any file. But, by combining these data we can come up with many other possible combinations that would be mappable. These are:

1 - the absolute count of persons, 1980
2 - the absolute count of persons, 1985
3 - absolute population change, 1980 to 1985
4 - relative population change, 1980 to 1985 as a %
5 - population density in 1980
6 - population density in 1985
7 - absolute change in population density, 1980 to 1985
8 - relative change in population density, 1980 to 1985 as a %

With these 3 basic variables we could also examine the relationship between population growth and population size. If we find that there is a significant relationship, then it would be possible to map the residuals from regression to see which areas fit the model and which areas show significant departures. There are many indices that can be derived from just these three variables. Add qualifiers of age, sex, race, migration, education levels, and income figures and the possibilities grow ever more cumbersome.

Now, where such data are available the burden is on the user to select which values should be read and to derive any combinations. Likewise, the burden is on the user to select a title, particularly when new indices are derived. The problem as we see it is to capture that reasoning, and dialog, that the expert would use to come up with a particular map or series of maps given a specific concern and a large data base. It seems to be an iterative process, where a person starts with one map and moves on to alternative maps representing the same data. Looking at an alternative may open a new avenue of exploration or it may take the inquirer back to the original or on to another variable or combination of variables. But, somewhere in the collection of data, assuming appropriate scales of analysis, there is normally information that if properly presented would help the decision maker make a more rational decision. By the same reasoning, it is probably true that shown the wrong data or even a bad representation of some relevant data, the decision maker may be mislead into making a less rational decision.

CONCLUSION

This developmental system has served many users. It has been a good training ground where students could get their hands on some hardware that had enough power to take them beyond their previous experiences. With each new development, another set of possibilities are opened up. Another student comes along and takes the system another step in another direction. And, in between these learning sessions, the system has been used to generate some effective slides for presentations.

REFERENCES

Kimerling, A. Jon, 1985, "Simultaneous Contrast and the Unclassed Choropleth Map," Technical Papers, American Congress on Surveying and Mapping, 45th Anual Meeting, 355.

Kimerling, A. Jon, 1986, Personal conversation.

Pfefferkorn, C., D. Burr, D. Harrison, B. Heckman, C. Oresky, and J. Rothermel, 1985, "ACES: A Cartographic Expert System," Proceedings, Auto-Carto 7, 399-407.

Robinson, Gary, and Michael Jackson, 1985, "Expert Systems in Map Design", Proceedings, Auto-Carto 7, 430-39.

Smith, Karen F., 1986, Robot at the Reference Desk, College & Research Libraries, Sept. 1986, 486-490.

Tobler, Waldo R., 1973, "Choropleth Maps Without Class Intervals?" Geographical Analysis 5:262-65.

COMPUTER-ASSISTED TERRAIN ANALYSIS ON A MICROCOMPUTER

Major Scott A. Loomer
Computer Graphics Laboratory
Department of Geography and Computer Science
United States Military Academy
West Point, New York 10996-1095
914-938-2063 or Autovon 688-2063

ABSTRACT

Prediction of cross-country movement rates for vehicles is one of several topics of considerable concern in both military and civilian applications. Algorithms have been developed by a number of organizations that predict speed based on vehicle characteristics and detailed terrain data. Sufficiently detailed terrain data are not readily available in digital form for most areas of interest. A microcomputer-based system has been developed for encoding existing analog (hardcopy) terrain data bases and processing them into a gridded data base. The databases thus created can be used to generate a number of cartographic products of which cross-country movement is one example. The system makes use of a low-cost, commercially available computer-aided design (CAD) program to provide both input and output to a wide variety of peripheral devices. Intermediate processing of the data is accomplished by custom programs developed by the author.

INTRODUCTION

Military Terrain Analysis

The primary function of the Army's field Terrain Analysis Detachments is to provide rapid response terrain analysis products to the field commanders. These products have been standardized by the Defense Mapping Agency (DMA) through the introduction of several procedural guides and an analog data base of terrain factor overlays. These overlays, known as tactical terrain analysis databases (TTADB), contain detailed information about the terrain with separate overlays for soil types, slopes, vegetation types, and others.

An example of a terrain product is the cross-country movement (CCM) map. Since a vehicle's capability to move off-road is influenced by several factors, most notably vegetation, slope and soil type, manual preparation is very tedious. An officer with recent service in a field terrain unit has estimated that while approximately 50% of the production effort made use of the terrain factor overlays, the terrain teams did not usually have the time required to create some of the more complicated terrain analysis products. Tactical terrain analysis is a prime candidate for automation. DMA is working toward the automation of topographic terrain analysis data, but it will be several years before such data are available for wide areas.

System Concept

The goal of the project described in this paper, named DigiTAS for Digital Terrain Analysis System, was to develop a low cost system of off-the-shelf hardware and internally developed software in order to automate the creation of standard DMA terrain products. Unlike previous programs to introduce automation to terrain units in the field, the proposed system would be capable of directly supporting the units' day-to-day mission. Further, the system is based on two separate modules, the first of which allows field creation of a terrain data base and the second of which manipulates that data. When DMA fields a digital terrain data base, the first module will become somewhat superfluous, but the second, with slight modifications, will be able to handle DMA produced data.

The algorithms for creating the terrain products have been well documented and are not presented here. The thrust of this project, as with much of the effort in automating cartography, is not in manipulating the data but in acquiring it to begin with. This paper deals with the design and development of a specialized terrain analysis system.

SYSTEM DESIGN CONSIDERATIONS

System Functions

Four functional requirements were identified for the DigiTAS system:

1. Digitize currently available DMA produced terrain factor overlays, converting the analog data into a gridded digital data base compatible with existing digital terrain elevation data (DTED).

2. Store digital data in as compact a form as possible, making use of data compression techniques and bit-mapping of data. Desired goal was to have all data for a 1:50,000 topo sheet on a single 1.2 MB floppy disk.

3. Support interactive manipulation of the data using algorithms that automate the DMA procedural guides to create DMA standardized terrain analysis overlays (e.g., winter concealment, cross-country movement).

4. Create hardcopy output overlays to existing map sheets.

System Hardware Configuration

The hardware configuration developed for the system is shown in Figure 1. All components of the system are readily available commercial products consisting of the following:

1. Microcomputer. An MS-DOS based microcomputer with high resolution graphics display. The programs developed support several types of input and output hardware, recognizing and making use of the maximum capabilities of the host system (e.g., high resolution displays, math co-processors, extended/expanded memory).

2. Input. A digitizing tablet to allow encoding of the terrain factor overlays. Different size digitizers are supported with paste functions allowing a small (12" x 12") pad to be used for digitizing an overlay in sections.

3. Output. A printer or plotter for output. Several types of output devices are supported from small format dot-matrix printers to large format plotters. All are capable of generating output which can yield the standard products, albeit some output may require limited manual drafting.

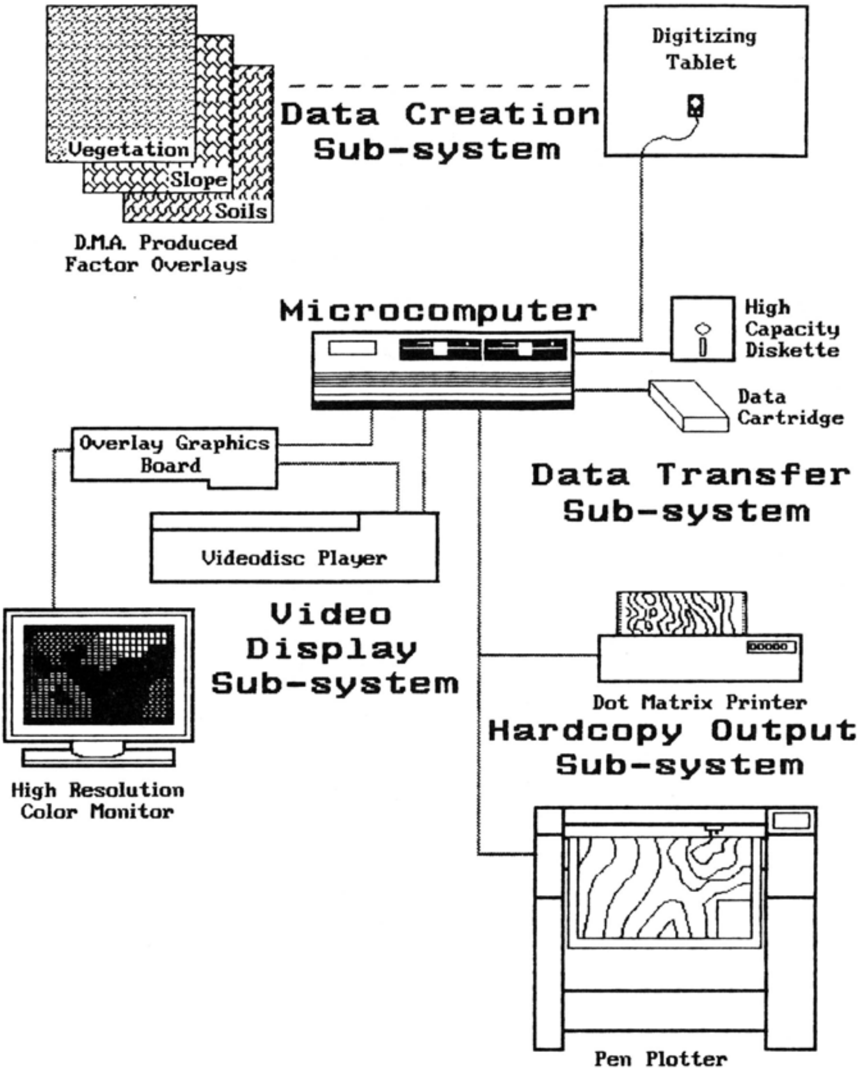


Figure 1: System Hardware Configuration

Specific Programming Requirements

At the start of the project, the specific programming requirements were identified:

- support for general purpose digitizing
- conversion of digitized vector data to a gridded system
- transformation of the data into the DTED-compatible coordinate system
- pasting together individually digitized segments of the terrain factor overlays
- bit-mapping the gridded data in as compact a form as possible
- implementing the DMA procedural guides as programming algorithms
- implementing device-independent display routines
- creating multiple output device drivers

To avoid the major task of developing the device-independent input/output drivers, the use of a commercial computer-aided design (CAD) program was investigated. Many low-cost high-performance CAD packages have recently entered the marketplace. Several were examined and most proved suitable for providing digitizing and editing support for encoding the terrain factor overlays. In addition, the programs could also handle output of the final products on pen plotters and pieced dot-matrix printers. Importing and exporting data to and from the CAD environment is usually accomplished via a translation program.

General Scheme of Operation

The data flow through the system is illustrated in Figure 2. The sequence is:

1. DMA produced terrain factor overlays are digitized. The digitizing scheme is the "spaghetti" approach with each line segment being digitized once.

2. The vector "spaghetti" generated by the digitizer is processed by a conversion program that accomplishes three tasks in sequence: building the line segments into polygons, tagging the polygons with the feature attribute code, and converting the polygons to a gridded database.

3. Depending on the standardized product desired, the data base is manipulated to provide an on-screen representation of the terrain product. If hardcopy is desired, the terrain data processing program converts the gridded overlay to a vector representation in the exchange format for the CAD program.

4. The CAD program generates plotter output on high end systems with full-size registered overlays. Low end systems can generate a series of true-scale sectionalized dot-matrix printouts which can be registered and traced to produce a full-size overlay.

Items 1. and 2. above constitute Module I of the system which will ultimately be superceded by DMA produced digital terrain analysis data.

Items 3. and 4. above constitute Module II of the system which can be modified easily to support DMA supplied digital terrain analysis data when it is defined and available.

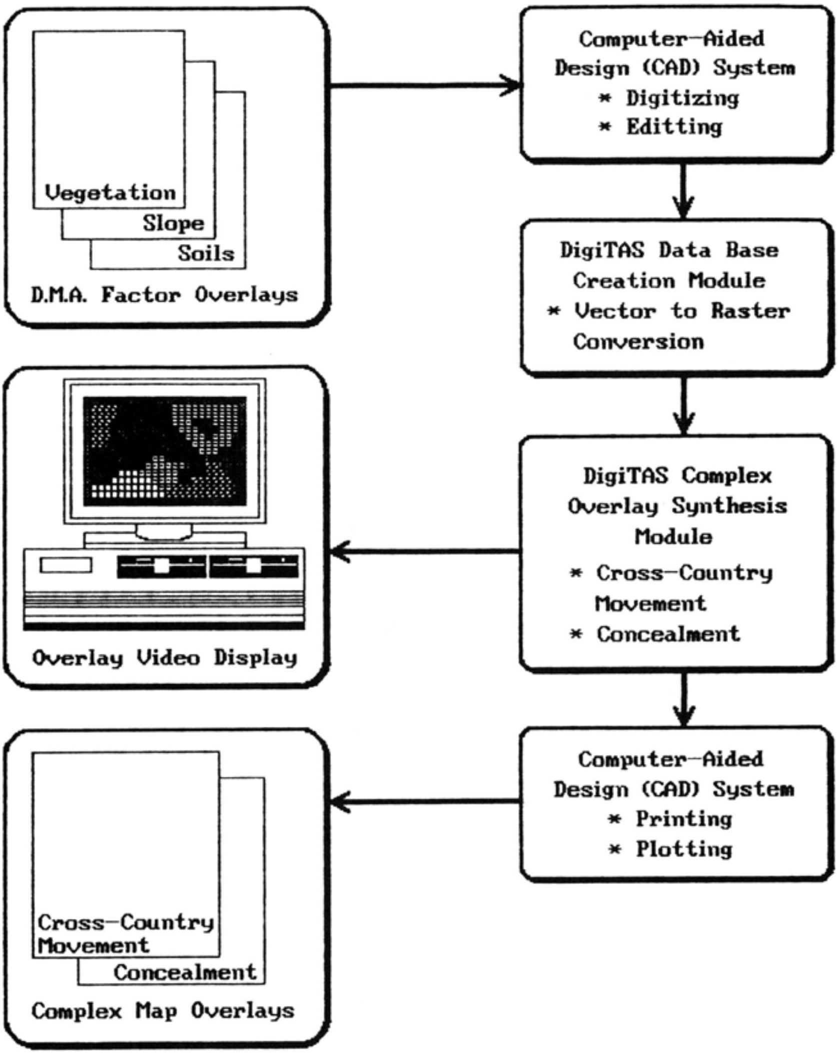


Figure 2: Data Flow through DigiTAS

DigiTAS Data Structure

Two approaches were considered in selecting the DigiTAS data structure, the first being to design a data structure optimized specifically for DigiTAS and the second being to base the design on the proposed DMA terrain data structure. The second approach was adopted to make the system as compatible as possible with the proposed DMA product as well as making interim use of data derived from DMA's terrain factor overlays. The design of the data structure for DigiTAS is therefore driven by several goals:

- retain all information content of the planning and tactical terrain analysis overlays that are digitized
- adhere to the proposed tactical terrain data (TTD) specifications to allow data to be easily subsetted from TTD when available
- balance the tradeoff between data compression and data manipulation

There are two basic components of the issue of data structure: data record structure (data content per unit area) and database structure (unit area content per cell). These are essentially independent of one another and are addressed below.

Data Record (Unit Area)

	# of Categories		# of Bits
Factor	TTADB	TTD	DigiTAS
1. Vegetation			
a. type	24	26	5
canopy closure			
b. summer	5	5	3
c. winter	N/A	5	3
d. height	10*	11	4
e. stem diameter	LUT	13	4
f. stem spacing	LUT	15	4
g. roughness	LUT	11	4
h. undergrowth	2	2	1
i. misc. other	N/A	?	0
2. Surface Configuration	8	9	4
3. Surface Materials			
a. type	20	21	5
b. state of ground	3	5	3
c. depth of material	2	3	2
d. surface roughness	5(LUT)	21	5
e. misc. other	N/A	?	0
Total bits to encode			47

* LUT - factors determined from a look up table

Table 1: Data Content of Terrain Databases

The DigiTAS encoding scheme aligns the categories described above as closely as possible with individual bytes in the data record to facilitate access to any specific item while keeping the data as compact as possible:

```
byte:      |    1    |    2     |    3    |    4    |
bit:       |01234 567|012 3 4567|0123 4567|0123 4567|
category:  | 1.a.|1.b|1.c|h|1.d.|1.e.|1.f.|1.g.| 2.  |
           |              Vegetation          |Surf|
                                               Conf
byte:      |    5    |    6     |
bit:       |01234 567|01 23456 7|
category:  | 3.a.|3.b|3c| 3.d.| |
           | Surface Materials|
```

Table 2: DigiTAS Bit-mapped Data Record

Database Structure

Currently, DMA-produced digital data comes in two distinctly different forms: gridded elevation data (DTED) and polygonal feature data (DFAD). The question of gridded versus polygonal data structures is well-argued and has vociferous supporters on each side. For DigiTAS, the decision was made to use gridded data for computational efficiency on the microcomputer host. The basic tradeoff is a larger mass storage requirement for gridded data but simpler processing of algorithms requiring determination of the intersection between various terrain factors. In the last decade, microprocessor power has improved by about two orders of magnitude from a 1MHz 8080 to the 16MHz 80386. In that same period, microcomputer mass storage has gone from 80 KB floppy disks to 500 MB write once read mostly (WORM) drives, an increase of almost 4 orders of magnitude. Therefore, the decision was made to opt for ease of processing over storage volume.

The next design consideration was the choice of reference grid and grid interval. Several possibilities were examined: registration to World Geographic System (WGS) coordinates, local spheroid geographic coordinates or the military standard Universal Transverse Mercator grid system. Each of the choices has advantages and disadvantages:

1. Latitude/longitude grid for 15' X 15' WGS cell
   advantages:
   - matches DTED data location
   - world-wide coverage without edge matching problems

   disadvantages:
   - difficult to register to base map or factor overlays
   - multiple cells to cover one map sheet

2. Latitude/longitude grid keyed to 15' x 15' base map
   advantages:
   - easy registration to base map or factor overlays for both input and output
   - one cell (or integral number of cells) per base map

   disadvantages:
   - requires transformation of WGS-based DTED to local datum

3. UTM grid keyed to base map
   advantages:
   - constant size unit area on the ground
   - precisely aligned with UTM grid

   disadvantages:
   - usually skewed with respect to map neatlines
   - considerable matching problems at edges of grid zone
   - requires transformation of WGS-based DTED to local datum

The current DMA produced gridded data (DTED) is referenced to the WGS at a geographic interval (3 arc seconds for low latitudes). The source materials for the DigiTAS data, the terrain factor overlays, are registered to base map sheets on a local spheroid. Since the desired end products are complex overlays keyed to base maps, the grid reference chosen for DigiTAS is registered to the base maps. The grid interval adopted matches that of DTED, 3 arc seconds of latitude and longitude at low latitudes shifting to 6 arc seconds of longitude by 3 arc seconds of latitude at higher latitudes. A simple local datum transformation and interpolation would allow DTED to be merged with DigiTAS data.

The final consideration was total data cell size. Since the unit area coding scheme generates 6 bytes per unit and there are 90,000 units per nominal 1:50,000 map sheet (15' by 15'), the DigiTAS data for each cell is approximately 540,000 bytes in size. This is an unfortunate size as it will neither fit on a MS-DOS standard floppy disk (360 KB) or in the microcomputer's RAM. This can be resolved by breaking the data cell into several files by either factor category or area. Since vegetation data alone account for 3-1/2 bytes, the only breakdown by factor category would be vegetation/surface configuration in one file (360 KB) and surface material in a second file (180 KB). Splitting the data in this fashion complicates the synthesis of certain terrain products that are dependent on all factors.

Alternatively, the file could be broken into subareas. Although it would only be necessary to divide the file in two to allow it to fit into memory or on a floppy disk, a more practical approach is to use a 5' by 5' cell size. This allows easy registration to the base maps as the internal 5' intersections are marked. Each cell is 60,000 bytes in size; coverage for a standard map sheet can be disseminated as 6 cells on one disk and 3 cells plus any miscellaneous files on a second disk or all files on one high-density disk. This is the structure that has been adopted for DigiTAS.

CONCLUSIONS

A microcomputer based terrain-analysis system offers a low-cost alternative to the manual methods currently employed by Army terrain analysis units. Creation of a terrain database from currently available terrain factor overlays is a practical task. Commercial computer-aided design software can be employed to provide input and output support and provide a standardized user interface for a wide variety of operations. Special purpose programs can transform the polygonal terrain factor data into a computationally more efficient gridded data structure. Interactive manipulation of the terrain database can produce standard terrain products such as cross-country movement maps in near real-time.

RESULTS OF THE DANE COUNTY LAND RECORDS PROJECT: IMPLICATIONS FOR CONSERVATION PLANNING

Bernard J. Niemann, Jr., and Jerome G. Sullivan
Land Information and Computer Graphics Facility
College of Agricultural and Life Sciences, School of Natural Resources
University of Wisconsin-Madison 53706

ABSTRACT

This paper presents the results of the Dane County Land Records Project, a four year cooperative research venture involving numerous local, state, and federal agency cooperators. The project has developed, tested, and evaluated a concept for a multipurpose land information system. Components of this concept have included reliance on individual data layers maintained by legislatively mandated agencies, and a common mathematical reference system to permit integration of the layers. Results of time and cost comparisons for manual digitizing and automated scanning, for data collection such as agricultural land use detection, and for landscape analyses are presented. Experiments with satellite geopositioning (Doppler Surveying and Global Positioning System), and inertial surveying methods are discussed. Implications for institutions using cooperative agreements are discussed and implementation principles are presented.

LEGISLATIVE MANDATE

For many years, the limitations of mapping technology have set the limits of the information available for the management of the land. The law might require decisions to be carried out with certain information. Planners and other land managers have resorted to the rationale that a plan is based on "best available information". One of the missing elements has historically been ownership information, or an identification of those impacted by planning proposals or those responsible for negative impacts to the environment. In the predigital period, it was possible to avoid the ownership record due to technical limitations that will not apply in the new technology.

Soil Erosion and Conservation Planning in Wisconsin

The case of soil erosion planning in Wisconsin provides an example of the evolution of an environmental management program. In a very few years, soil conservation has moved from an isolated provision of technical assistance for willing farmers to a quasi-regulatory program integrated with many other programs. Information technology has not yet played a direct role in this process. Soil conservation became a national issue over firty years ago during the dustbowl era. Despite substantial efforts, soil erosion is still a major problem.

Wisconsin has taken the approach of incorporating the conservation districts directly into the organization of Wisconsin government at the state and county level (Arts, 1982; 1984). The state has also created a new program with the intention of reducing soil loss. Some district staff see this as a simple continuation of past policies and procedures, but there are some fundamental shifts in the information requirements. The new program is described in Chapter 92 of State Statutes (dated 1981) and implemented in Administrative Rule Ag 160 (dated 1984). The statute gives an overall description of the plan:

> *Each land conservation committee shall prepare a soil erosion control plan which does all of the following: ...*
> *2. Identifies the parcels and locations of the parcels where soil erosion standards are not being met. ... [92.10 (5)a]*

The administrative rule specifies the program goals in greater detail:

> *The goal of the soil erosion control program is to reduce soil erosion caused by wind or water on all cropland in Wisconsin to T-value by the year 2000. T-value means the maximum average annual rate of soil erosion for each soil type (specificed in the SCS Technical Guide) [Ag160.03 (16)]*
>
> *For watersheds or other cropland areas determined by the land conservation committee to be of highest priority, the soil erosion control plan shall include detailed estimates of cropland erosion rates. Estimates shall be sufficiently detailed to permit the identification of individual parcels of cropland which are in need of erosion control practices. [Ag 160.05 (4b)] (emphasis added)*

Cross-Compliance in Wisconsin

After this structure for the soil erosion planning process was put in place, the need for integrated land information was increased by further action at the state level. A major program in the state budget is Farmland Preservation, which provides a state income tax credit for payments of local property tax on agricultural parcels. In return for the tax credit, the farmer must keep the land in agricultural use, enforced by either zoning or contract with the state. In the state budget recently adopted, there is a mandate to integrate farmland preservation with soil conservation., Under the new scheme, a farmer will have to provide a certificate from the county Land Conservation Committee showing compliance with soil erosion standards before the zoning administrator can certify the farmer's tax credit. This requirement could not have been anticipated from a user needs assessment, but the intention to integrate information on resources and parcels was already leading in this direction (see also Sullivan et al., 1984, 1985).

Cross-Compliance in the Federal Farm Bill

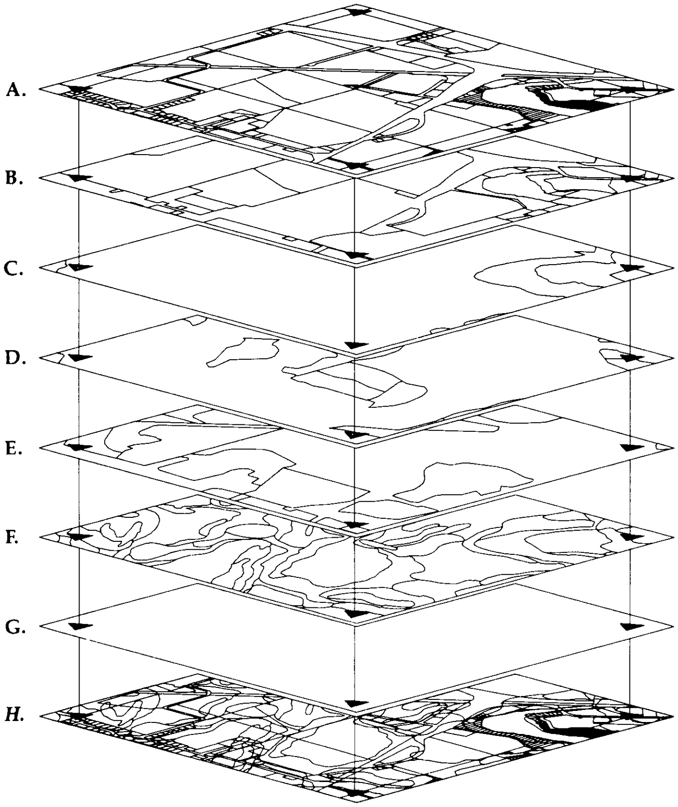
The 1985 Federal Farm Act also includes provisions to restrict poor management of marginal farmlands: "Sodbuster" for the provision adressing highly erodible lands, and "Swampbuster" for the provision addressing drainage of wetlands. These provisions actually require the conservation districts (at the county level) to integrate the resource information on soil capability with the information on owners and land users who receive any farm subsidy. Because the integration which Congress intends is similar to the Wisconsin case, we believe that our study in Dane County provides an adequate demonstration that a multipurpose land information system can efficiently and equitably respond to these requirements.

ORDERS OF MAGNITUDE: GEOPROCESSING, GEOPOSITIONING, REMOTE SENSING

The central components of the DCLRP concept involve the maintenance of individual data layers in a digital form by the agencies mandated with their generation, and the use of a mathematical reference framework for linking individual layers (Chrisman et al., 1984; Chrisman and Niemann, 1985) (see Figure 1). In achieving these goals, the Dane County Land Records Project has utilized advanced geoprocessing software to perform topological polygon overlay (ODYSSEY), and has investigated advanced geopositioning technologies (Doppler, Inertial, Global Positioning System). The DCLRP has also incorporated classified digital remotely sensed imagery through a vectorization process (Ventura et al., 1985, 1986). In the implementation and use of a multipurpose land information system, it appears that order of magnitude efficiencies are possible in geoprocessing, geopositioning and use of remote sensing.

Advanced Geoprocessing Software

Manual digitizing of mylar soil sheets (1:15840, 7 square miles, average 300 polygons), combined with editing time (including automated error checking) to produce a topologically clean sheet, averaged 12 hours (Chrisman, 1986c; Ujke, 1984). The adoption of scanning digitizing (Chrisman, 1986a) was found to reduce combined digitizing and editing time to 4 hours. A photogrammetric technique for removing relief distortion from rectified photobases, as in the case of the SCS soil sheets, using USGS digital elevation models (DEM) was developed by the DCLRP (Barnes, 1984,



Concept for a Multipurpose Land Information System

Section 22, T8N, R9E, Town of Westport, Dane County, Wisconsin

	Data Layers:	**Responsible Agency:**
A.	Parcels	Surveyor, Dane County Land Regulation and Records Department
B.	Zoning	Zoning Administrator, Dane County Land Regulation and Records Department
C.	Floodplains	Zoning Administrator, Dane County Land Regulation and Records Department
D.	Wetlands	Wisconsin Department of Natural Resources
E.	Land Cover	Dane County Land Conservation Committee.
F.	Soils	United States Department of Agriculture, Soil Conservation Service.
G.	Reference Framework	Public Land Survey System corners with geodetic coordinates
H.	*Composite Overlay*	*Layers integrated as needed, example shows parcels, soils and reference framework*

Land Information and Computer Graphics Facility,
College of Agricultural and Life Sciences, School of Natural Resources
UNIVERSITY OF WISCONSIN-MADISON

Support provided by the Gordon H Baker Fund of the University of Wisconsin Foundation

1985,1986). A "Zipping" process was developed to automate the edgematching of separately compiled map sheets, using an approach which limits calculations to edges of the maps (Beard and Chrisman, 1986).

Satellite Geopositioning

The individual layers which were brought together were transformed to state plane coordinates (SPC) using section corners and quarter corners for control. Establishment of SPC for the PLSS monuments involved a comparison of traditional manual surveying techniques and satellite geopositioning technologies (Vonderohe, 1984a,b; Vonderohe and Mezera, 1984; Vonderohe et al., 1985; von Meyer, 1984a,b; von Meyer et al., 1985). Our research has demonstrated an order of magnitude difference in both time and cost for these methods of establishing the reference framework. Whereas manual surveying methods required several days and $1000's to establish coordinates for a point, Doppler satellite methods required only two days and $100's, and global positioning system (GPS) methods required only hours and $100's.

Remote Sensing and Digital Image Processing

In conjunction with the University of Wisconsin Environmental Remote Sensing Center, the DCLRP acquired, classified, and vectorized Landsat Thematic Mapper data for agricultural lands in Dane County (Ventura et al., 1985, 1986). Again, an order of magnitude difference was found between the 1/2 hour per PLSS section required for traditional manual photointerpretation of Agricultural Stabilization and Conservation Service (ASCS) 35 mm slides and compilation on the SCS soil photobase, versus minutes per PLSS section to perform the digital classification.

ASSESSING SOIL EROSION POTENTIAL FOR EACH LANDOWNER

The process used for determining soil erosion potential involved an application and automation of the Universal Soil Loss Equation (USLE) for agricultural parcels (Wischmeier and Smith, 1965), as prescribed in Administrative Rule Ag 160.

The accompanying maps portray this application for the Town of Oregon, Dane County, Wisconsin, T5N, R9E. Figure 2 was produced by manually digitizing 36

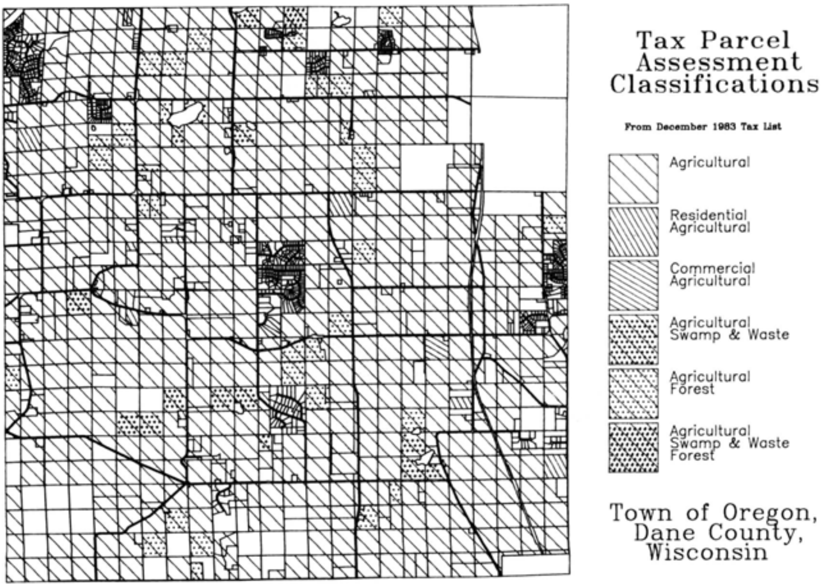

Tax Parcel Assessment Classifications
From December 1983 Tax List
Agricultural
Residential Agricultural
Commercial Agricultural
Agricultural Swamp & Waste
Agricultural Forest
Agricultural Swamp & Waste Forest
Town of Oregon, Dane County, Wisconsin

Figure 2: Tax Parcel Assessment Classifications, Oregon Twp., WI

section maps of tax parcels maintained by the County Surveyor at 1:4800, most on linen bases. After editing and edgematching, each tax parcel ploygon was assigned its unique identifier, as recorded on the County Zoning Administrator's section maps.
The identifier permitted access to the tax parcel assessment classifications recorded in the automated tax rolls of the County Tax Lister. Only those parcels having an agricultural assessment classification are shaded on this map; areas with classifications other than agricultural, swamp and waste, or forest were excluded from the study.

The use of an automated system for overlay and analysis of map layers, has provided the County Land Conservation staff with a workable tool for prioritizing their field observations and landowner contacts as they work to implement the soil erosion control plan. Whereas before a manual overlay analysis for a township might take days, the same analysis can now be performed in an hour. Similarly, a manual interpretation of an individual farm's eligibility for a given program might have required hours; the computer assisted interpretation requires only minutes per farm. Through development of automated case files and linkage to the digital layers of land information, the county land conservation staff is moving toward a system for monitoring compliance.

This process of automating existing land records such as ownership, soils and agricultural use and applying the USLE has been demonstrated elsewhere (see Chrisman et al., 1986a,b). The maps on the following pages demonstrate the application of this process. Figure 3 illustrates which parcels and landowners will not be in compilance (A > 2T and T < A < 2T) without employing some additional conservation management procedures.

Figure 4 illustrates the impact to soil erosion by employing conservation tillage practices: all parcels are brought below the level of 2T, and many of those with moderate erosion potential are brought within the acceptabel level and no longer exceed tolerable soil loss.

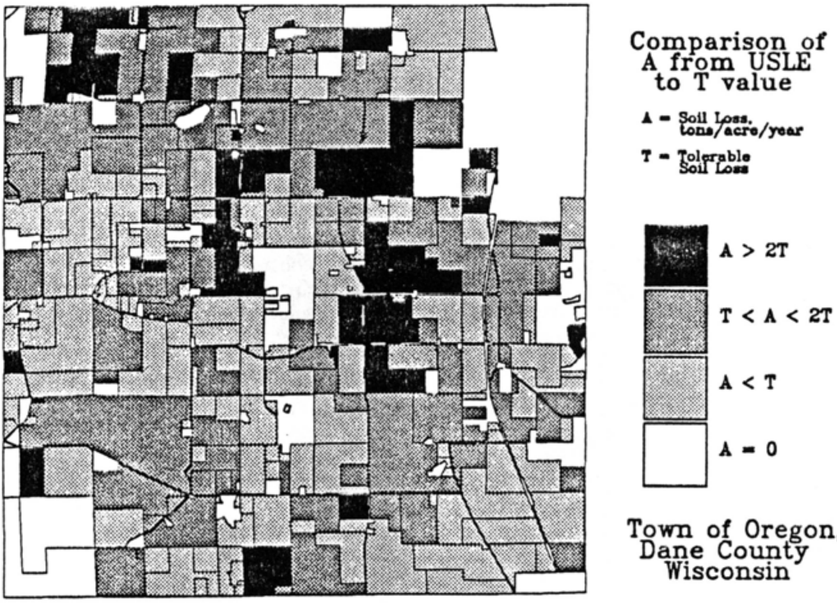

Comparison of A from USLE to T value
A = Soil Loss, tons/acre/year
T = Tolerable Soil Loss
A > 2T
T < A < 2T
A < T
A = 0
Town of Oregon, Dane County Wisconsin

Figure 3: Comparison of A from USLE to T value, Oregon Twp., WI

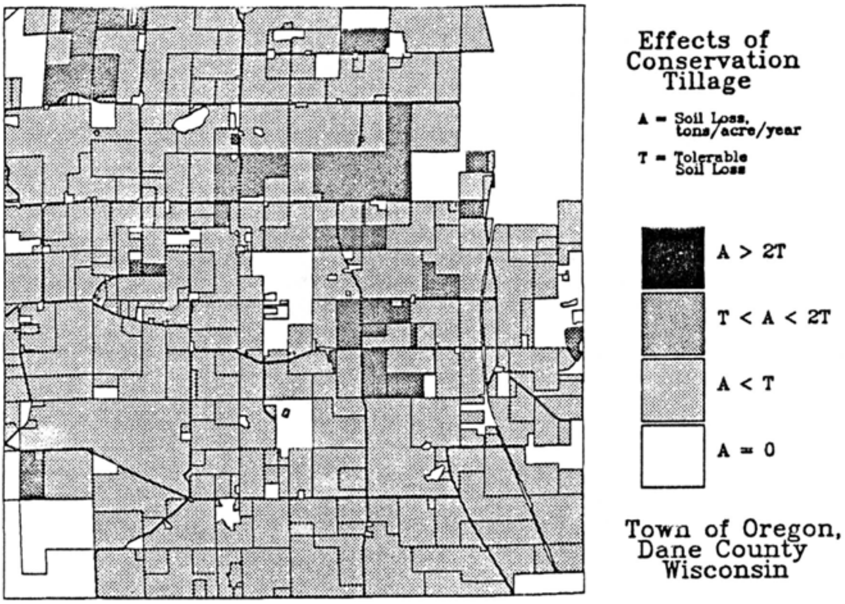

Effects of Conservation Tillage
A = Soil Loss, tons/acre/year
T = Tolerable Soil Loss
A > 2T
T < A < 2T
A < T
A = 0
Town of Oregon, Dane County Wisconsin

Figure 4: Effects of Conservation Tillage, Oregon Twp., WI

MODERNIZATION PRINCIPLES DERIVED FROM EXPERIENCES GAINED DURING THE DANE COUNTY LAND RECORDS PROJECT

As a result of the project, a number of social, economic, institutional, and technological trends have been identified in the process of addressing modernization issues. Inititally, taking advantage of new land records and information technology requires educational and institutional changes. In bringing about modernization, the following principles for the development and implementation of modern, multipurpose land information systems need attention.

Automation
A system neeeds to be based upon intelligent concepts such as topological vector data structures, in which: spatial locations, attributes, and their relationships (ie, adjacency and connectivity) can be maintained; logical and spatial search as can be conducted; and cross-checking of consistency, closure, and unique identification of areas and attributes are possible.

A system needs to support analytical capabilities such as topological polygon overlay, network analysis, buffer generation, etc.

A system needs to accomodate data capture and conversion (ie. raster to vector and vector to raster) between diverse routine and non-routine land record sources.

Geopositioning
A system needs to be constructed upon a geodetic reference framework.

A system needs to be based upon remonumentation and determination of coordinates for

Public Land Survey System (PLSS) corners and other survey monuments to provide both a spatial reference system and an improved legal system for property description.

Standards for geopositioning need to be established.

Applications

A system needs to be multi-layered, including property descriptors, tax assessment parcel records, and unique parcel identifiers to assure multiple use applications.

A system implementation needs to include a pilot project to test and demonstrate applications. High use, high visibility applications, should be chosen and output examples should be provided early on.

Quality

System evolution needs to include the development and adoption of standards for the various records, including property, resource mapping, remonumentation, and geodetic control.

A system needs to include procedures which clearly documents the source, lineage (original scale, accuracy) and method of automation for each record to assure logical consistency and completeness.

A system needs to automate records at the greatest available detail to assure non-degradation of original positional and attribute accuracy, and therafter perform aggregations for more general applications.

Institutional

System implementation should focus initially on institutional cooperation before addressing technical issues.

System implementation should determine short-term and long-term custodial mandates and maintain responsibilities for each land record in the system.

System implementors must recognize that implementation is a long-term venture and an investment, and will require continuous evaluation rather than merely a one-time experience.

System advocates need to recognize that the approach is interdisciplinary and therefore need to involve a variety of disciplines and professionals in the initial system development and planning stages.

System designers need to insure that the records base is unbiased, politically and institutionally neutral, to assure its broad official and private use for both daily management and policy making functions.

Economics

System designers, implementors and users need to ensure that the record base allows for efficient, yet comprehensive and exhaustive analysis to ensure fair and equitable treatment to all.

System implementors need to incorporate new technologies into the operating system, such as the global positioning system and scanning technologies, to gain needed efficiencies in geopositioning and digital conversion of land records.

System implementors need to recognize that some applications and analyses will be accomplished much faster then formerly, while other applications which were not possible will emerge, and unanticipated benefits will result.

System implementors and managers need to recognize that a learning curve exists in system development and use. Initial costs to convert and use records will be higher than for these same activities after experience with the system has been gained.

In summary, system advocates, implementors, managers and those responsible for overall approval need to recognize that operating efficiencies will result; that timeleness will improve in that analyses can be accomplished faster; that synergism will result in being able to do things that were impossible manually; and that analyses can be accomplished comprehensively and exhaustively, resulting in fair and equitable treatment of all.

SUMMARY: TECHNICAL AND INSTITUTIONAL FEASIBILITY OF LIS

In this evaluation of a multipurpose land information system we have demonstrated that it is technically feasible to identify land ownership parcels where soil erosion standards are not being met. We have demonstrated the utility of combining advanced geoprocessing, geopositioning, and remote sensing technologies. We have also demonstrated the need for flexible data structures, such as layering, and analytical procedures, such as topological polygon overlay, to respond to new land management questions and mandates such as cross-compliance. We have also documented what appears to be institutional interest in these issues.

There is legislative interest both at the state and national level to ensure that society receives equitable returns upon public investments in support of agriculture. As a result, farm supports of various kinds are being linked to reduction in soil erosion and minimization of wetlands destruction. As the public awareness of land management becomes linked to broader concerns, there will be increased needs to integrate diverse information, such as the natural resource and ownership layers used for the Wisconsin soil erosion plan. The same tools and procedures which are essential to multipurpose land information systems may be those mechanisms which stewards of the land need to implement land management programs.

It is possible that these technologies will have the same social impact as the automation of the Census had upon the implementation of racial desegregation in the U.S. The ability to establish defensible indices of segregation, based upon manipulation of the automated Census records, formed the information base that made desegregation an achievable goal. With the advent of modern information concepts and technologies which allow for merger of various records sets, are we at the brink of such an impact on rural land management? Will the application of such technologies provide for sufficient certainty to allow legislative mandate of land management programs implemented at the parcel level? If so, this could have profound impacts on those who own and manage rural America.

REFERENCES

Arts, J. 1984. Coordinating Soil Conservation Programs: The Wisconsin Approach. *J. Soil and Water Conservation*, Nov. -Dec. 1984, p.354-356.

Arts, J. 1982. *Recent Amendments to Wisconsin's Soil and Water Conservation Law: Their Purpose and Anticipated Effects*, Department of Agriculture, Trade, and Consumer Protection, Madison, WI, 10p. + appendices.

Barnes, G. 1984. Minimization of Photo Distortion in Soils Sheets: A Progress Report for the Dane County Land Records Project. Unpublished, DCLRP, UW-Madison, 9 p.

Barnes, G. 1986. Report on Software Developed to Minimize Photographic Distortion. Unpublished, DCLRP, UW-Madison, 16p.

Barnes, G., and A.P. Vonderohe. 1985. An Analytical Technique to Account for Photographic Distortions in Natural Resource Records. *Proc. URISA* , Denver, 1:171 -180.

Beard, M.K., and N.R. Chrisman. 1986. Zipping: New Software for Merging Map

Sheets, *Proc. ACSM* , 1:153-161.

Chrisman, N.R. 1986a. Effective Digitizing: Advances in Software and Hardware, *Proc. ACSM,* 1:162-171.

Chrisman, N.R. 1986b. Error in Overlaid Categorical Maps. Presentation at the Annual Meeting of the Association of American Geographers, Minneapolis.

Chrisman, N.R. 1986c. Quality Report for Dane County Soil Survey Digital Files. p.78-88 in H. Moellering (ed.), Report 7, NCDCDS, Columbus OH.

Chrisman, N.R. 1987. Accuracy of Map Overlays: A Reassessment. *Landscape Planning,* in press.

Chrisman, N.R., D.F. Mezera, D.D. Moyer, B.J. Niemann, J.G. Sullivan, A.P. Vonderohe. 1986a. Soil Erosion Planning in Wisconsin: An Application and Evaluation of a Multipurpose Land Information System. *Proc. ACSM*, Washington, DC, 3:240-249; also presented at FIG, Toronto, Ontario.

Chrisman, N.R., D.F. Mezera, D.D. Moyer, B.J. Niemann, J.G. Sullivan, A.P. Vonderohe. 1986b. Soil Erosion Planning in Wisconsin: Order of Magnitude Implications of Integrating New Technologies. *Proc. URISA* , Denver, 1:117-128.

Chrisman, N.R., D.F. Mezera, D.D. Moyer, B.J. Niemann, A.P. Vonderohe. 1984. *Modernization of Routine Land Records in Dane County, Wisconsin: Implications to Rural Landscape Assessment and Planning*. URISA Prof. Paper 84-1. 44p.

Chrisman, N.R., D.F. Mezera, D.D. Moyer, B.J. Niemann, A.P. Vonderohe, J. Sonza-Novera. 1985. Dane County Land Records Project. Poster Session, *Auto-Carto 7,* Washington, DC, p.512-513.

Chrisman, N.R., and B.J. Niemann. 1985. Alternative Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning. *Auto–Carto* 7, Washington, DC , p.84-94.

DCLRP. 1984. Wisconsin Land Information Packet Number 1: Satellite Positioning Technology. DCLRP, UW-Madison, 24p.

Green, J., and D.D. Moyer. 1985. Implementation Costs of a Multipurpose County Land Information System. *Proc. URISA* , Ottawa, 1:145-151.

Krohn, D.K. 1984. The least squares adjustment of Oregon Township. Unpublished, DCLRP, UW-Madison, 10p.

MacGaffey, N. 1985. Introduction to Homer Map Editing in Odyssey. Unpublished, Land Information & Computer Graphics Facility, UW-Madison, 25p.

Moyer, D.D. 1983. Decision Support Systems for Policy and Management. *Proc. URISA*, Atlanta, p.136-145.

Moyer, D.D., J. Portner, D.F. Mezera. 1984. Overview of a Survey-Based System for Improving Data Compatibility in Land Records Systems. *Computers, Environment and Urban Systems*, 7(4):349-358.

Niemann, B.J., N.R. Chrisman, A.P. Vonderohe, D.F. Mezera, D.D. Moyer. 1984. Dane County Land Records Project - Phase 1 Report. UW-Madison, 16 p.

Niemann, B.J., N.R. Chrisman, A.P. Vonderohe, D.F. Mezera,D.D. Moyer. 1984. Evaluate a Computerized Land Records System, Locally Responsive, Large Scale Format with Survey Monumentation: A Case Study of Westport, Dane County, WI. DCLRP, UW-Madison, 35p.

Niemann, B.J. and J. Portner. 1984. Computing Solutions at the Ownership Scale of Design. *Landscape Architecture*, 74(6):56-59.

Sullivan, J.G., B.J. Niemann, N.R. Chrisman, D.D. Moyer, A.P. Vonderohe, D.F. Mezera. 1984. Institutional Reform Before Automation: The Foundation for Modernizing Land Records Systems - A Case Study of Dane County, Wisconsin. Wisconsin Land Information Reports, UW-Madison, 17p.

Sullivan, J.G., B.J. Niemann, N.R. Chrisman, D.D. Moyer, A.P. Vonderohe, D.F. Mezera. 1985. Institutional Reform Before Automation: The Foundation for Modernizing Land Records Systems. *Proc. ACSM*, Washington, DC, p.116-125; also presented at the FIG International Symposium: The Decision Maker and Land Information, Edmonton, Alberta, p.383-391.

Sullivan, J.G., M.K. Beard, J. Sonza-Novera, N.R. Chrisman, B.J. Niemann. 1985. Multipurpose Land Information Systems: Institutional Innovations, Technological Trends, Implications for Landscape Architecture Education. *Proc. Council of Educators in Landscape Architecture*, Champaign-Urbana, IL, p.10-14.

Sullivan, J.G., and N.R. Chrisman. 1983. User's Documentation for Spaghetti Digitization and Related Software at the Land Information and Computer Graphics Facility. Unpublished, Land Information and Computer Graphics Facility, UW-Madison, 35p.

Sullivan, J.G., N.R. Chrisman, B.J. Niemann. 1985. Wastelands vs. Wetlands in Westport Township, Wisconsin. *Proc. URISA* , Ottawa, 1:73-85.

Ujke, J. 1984. Compilation Errors and the Correction Process. Unpublished, DCLRP, UW-Madison, 5p.

Ventura, S.J., N.R. Chrisman, A.P. Vonderohe. 1986. Quality Analysis of Digital Line Graph Data for the Dane County Land Records Project. Unpublished, DCLRP, UW-Madison, 22 p.

Ventura, S.J., B.J. Niemann, T.M. Lillesand. 1985. Land Cover for Soil Erosion Control Planning from Landsat Thematic Mapper Data. *Proc. URISA* , Ottawa, 1:86-95.

Ventura, S.J., J.G. Sullivan, N.R. Chirsman. 1986. Vectorization of Landsat TM Land Cover Classification Data. *Proc. URISA*, Denver, 1:129-140.

Vonderohe, A.P. 1984a. Global Positioning System Experiment: An Interim Report for the Dane County Land Records Project. Unpublished, DCLRP, UW-Madison, 27 p.

Vonderohe, A.P. 1984b. University of Wisconsin's Experience with GPS. Wisconsin Land Information Packet No. 1: Satellite Positioning Technology, UW-Madison, 2 p.

Vonderohe, A.P., and N.R. Chrisman. 1985. Tests to Establish the Quality of Digital Cartographic Data: Some Examples from the Dane County Land Records Project. *Auto-Carto 7*, Washington, DC, p.552-559.

Vonderohe, A.P., and D.F. Mezera. 1984. A Cooperative Venture: Dane County Doppler Survey. *Proc. ACSM*, San Antonio, TX, p.395-401.

Vonderohe, A.P., D.F. Mezera, and N.R. von Meyer. 1985. Geopositioning Experiences at the University of Wisconsin - Madison. *Proc. URISA*, Ottawa, 1:152-64.

von Meyer, N.R. 1984a. Report on the Southeast Quarter of Section 28, Oregon Township. Unpublished, DCLRP, UW-Madison, 20p.

von Meyer, N.R. 1984b. Westport Inertial Surveyor Summary of Coordinates and Trends. Unpublished, DCLRP, UW-Madison, 20p.

von Meyer, N.R., D.K. Krohn, D.F. Mezera, A.P. Vonderohe. 1985. Dane County Inertial Surveying Experiment. *Proc. ACSM*, Indianapolis, p.239-248.

Wischmeier, W.H., and D.D. Smith. 1965. *Predicting Rainfall Erosion Losses From Cropland East of the Rocky Mountains.* USDA Agricultural Research Service, 45p.

Wisconsin Administrative Rules, Chapter Ag 160: Soil Erosion Control program (1984).

Wisconsin Department of Agriculture, Trade, and Consumer Protection. 1984. *Soil Erosion Control Planning Manual.*. Madison, 92p.

Wisconsin Statutes, Chapter 92: Soil and Water Conservation (1981).

ACKNOWLEDGEMENTS

This work was performed under a cooperative research agreement among the University of Wisconsin-Madison, Dane County, Wisconsin Department of Natural Resources, Wisconsin Department of Agriculture, Trade and Consumer Protection, and the USDA Soil Conservation Service. Additional support and/or assistance was provided by UW-Madison Environmental Remote Sensing Center, Wisconsin's University Industrial Research Foundation, US Geological Survey, Bureau of Land Management, Wisconsin Department of Transportation, Wisconsin State Cartographer's Office, Madison Area Surveyor's Council, and Wisconsin Power and Light.

The co-principal investigators in the Dane County Land Records Project were N. Chrisman, D. Mezera, D. Moyer, B. Niemann, and A. Vonderohe. Special thanks to G. Barnes, K. Beard, K. Cotter, M. Dawood, C. Ghilani, J. Green, D. Krohn, J. LaVerriere, N. MacGaffey, G. Mahun, J. Sonza-Novera, J. Sullivan, M.Thoma, S. Ventura, and N. von Meyer for research assistance. Thanks also to cooperators K. Connors, R. Martin, M. Riggle, J. Ujke, and many others unmentioned.

THE DISPLAY OF BOUNDARY INFORMATION: A CHALLENGE IN MAP DESIGN IN AN AUTOMATED PRODUCTION SYSTEM

Dana Fairchild
Geography Division
U.S. Bureau of the Census
Washington, D.C. 20233

ABSTRACT

Political and statistical boundaries are among the most important elements displayed on maps produced by the Census Bureau. The use of a totally automated system for mapping from TIGER files presents some interesting challenges in the display of these boundaries, the most important of which involves the display of multiple coincident boundaries. This paper explores several alternative methods of displaying boundaries on maps produced in batch mode on a 200 dot-per-inch monochromatic electrostatic raster plotter. Examples of maps generated using the different methods and the results of a survey conducted to solicit user preferences are included.

BACKGROUND

Political and statistical boundaries are among the most important elements displayed on Census Bureau field maps. With the exception of maps used by Census Bureau enumerators to locate housing units in the field, the purpose of most census maps is to accurately depict the boundaries of areas for which data are or will be tabulated. Even on enumerator maps the display of the limits of the enumeration area is of critical importance. For some Census Bureau operations, the boundaries that are displayed are coincident with one another; in other words, they run along lines on the earth's surface defined by the same coordinates. Coincidence of two to four boundaries is very common; on some maps as many as eight boundaries may run along the same line.

There are several criteria involved with the display of boundary information that must be met in either a manual or automated map-making environment:

1. A wide audience of users with widely varied levels of experience in map reading must be able to interpret the types and locations of boundaries on a Census Bureau map.

2. Boundaries that run along earth base features, such as roads and streams, must not obscure these underlying features or the text that identifies them.

3. Boundaries that are coincident with one another must be displayed in such a manner that each individual boundary is identifiable while the coincidence itself is evident.

While meeting the above criteria presents complications regardless of whether manual or automated map production systems are used, there is a definite advantage to a manual system. Simply put, the individual creating the artwork can use his or her judgement in the placement of a boundary or boundaries, so that the correct boundary location can be inferred if plotting the exact locations cannot be achieved without creating visual clutter. For example, if three boundaries follow a road and plotting each of the boundaries over the road causes obliteration of the road and boundary symbology, one or more boundary symbols can be offset from the road just enough to reduce image congestion yet allow for the correct inference of the true boundary locations. This same result can be accomplished in an automated environment if an interactive review and edit of the graphic image occurs before map plotting. Offset boundaries can also be accomplished in a totally automated mode. The plotting program can determine when an offset is needed and perform the necessary changes in symbology locations; however, this is extremely costly in terms of processing time.

When multiple boundary symbols are coincident, solid symbols are more likely to obliterate underlying features and one or more of the boundaries than are screened symbols. Screened symbols are generally more versatile than solid symbols because the reader is able to see solid linework beneath the dot patterns. Still, when several boundaries are overplotted along the same line, the image can quickly darken and the necessary information is difficult to interpret, if it is not lost completely. Traditionally the offset approach discussed above is used, and the same advantages of human intervention apply.

Color is a tool that can be used to help solve this problem because it assists the reader in distinguishing between symbols or groups of symbols, but it is also very expensive and time-consuming to use. Because of these expenses, the Census Bureau will most likely restrict the use of color to publication-quality maps.

The Census Bureau must produce hundreds of thousands of maps for field use within an extremely tight time frame (6-9 months) for use in collection operations for the 1990 decennial census. All maps will be generated by computer in batch mode using TIGER Files*. Although color plotters may be used for some map types, black and white electrostatic raster plotters will be used for the production of field maps for data collection activities. At this time it is expected that 200 dot-per-inch (DPI) technology will be used. Thus, the following constraints are placed on the methodology used for displaying boundaries on Census Bureau field maps:

1. No use of color
2. 200 DPI electrostatic raster plotters for output
3. All maps generated in a totally automated environment (no interactive review and/or edit)

*See papers by Kinnear, Knott, and Meixler in these proceedings for discussions of the TIGER System.

The purpose of this paper is to explore several alternative methods for displaying boundary symbols within the constraints discussed above. Several methodologies are discussed and examples of portions of maps reflecting the different approaches are provided.

Because the ability of the map reader to interpret boundaries is the primary concern, a survey was conducted to learn of the preferences of different groups of users for one method over the others. The results of that survey are included.

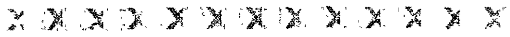
AUTOMATED BOUNDARY DISPLAY METHODS

Three approaches to the automated display of boundary symbols are discussed here. Each of the three uses symbols that are screened rather than solid because the plotting programs will place boundaries directly over base features when they follow them, rather than perform the complex and time-consuming calculations need to offset the boundaries.

Method 1 - Overplotted symbols
Each boundary is symbolized by a uniquely shaped screened symbol:

COUNTY AMERICAN INDIAN RESERVATION

When more than one boundary runs along the same line, the symbols are plotted over one another:

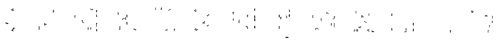


Screening is accomplished by a repeating 4 x 4 raster matrix pattern. The raster patterns that make up the individual symbols are unique for each symbol so that when one boundary overprints another, different rasters are turned on and the area of overlap is darkened. This is necessary for symbol shape distinction.

Darker matrix patterns allow for easier distinguishability of individual symbol shapes; however, overplotting several boundaries with dark patterns (for example, four rasters on in a 4 x 4 matrix) causes the area of focus to become too dark to easily identify each of the component boundaries. At the other end of the spectrum, a one-raster pattern is too light to define many shapes. Two rasters in a 4 x 4 matrix appears to work best for the overplotting method; more boundaries can be overlaid before approaching black. Even so, shape definition is not optimal in this format.

Method 2 - Alternating symbols
As with Method 1, each boundary is symbolized by a uniquely shaped screened symbol. Instead of overplotting coincident boundaries, the shapes of the component coincident boundary symbols alternate along the

boundary line. An example of the same coincident boundaries used to illustrate Method 1 would be:



With this method, darker raster patterns can be used for enhanced shape distinction because only base features are overplotted – not other boundary symbols; however, more linear space is needed for alternating the symbols. The symbols themselves cannot be smaller than .10" without losing shape on a 200 DPI device and more complex shapes cannot be acceptably defined at that size. Assuming a .15" size with .10 inch spacing between shapes, four shapes (boundaries) can be shown in one map inch. Sometimes boundaries are coincident for less than one map inch and all component boundaries cannot be symbolized in the allotted space using this method.

This is comparable to the problem encountered with Method 1 when so many boundaries occupy the same line that the area becomes too dark to decipher the component boundaries. With both methods, the map reader will have to infer which boundaries are coincident. These confusing situations should not occur frequently and the map reader should be assisted in his or her inference by locating and identifying the individual boundaries leading into the problem area.

Method 3 - Unique multiple boundary symbol
The same symbol appears on the map whenever two or more boundaries are coincident. A key number appears next to the symbol and also in the map legend, where the component boundaries are identified. This method eliminates the problem of unacceptable dot density and inadequate amount of space for correct boundary display. The drawback is that it causes the map reader to be totally dependent on the map legend since individual symbols are not uniquely symbolized as part of the multiple boundary. Example:

LEGEND

MULTIPLE BOUNDARIES

$\boxed{2}$ COUNTY AND AMERICAN INDIAN RESERVATION

Boundary Hierarchy
Certain political and statistical areas nest within others. For example, counties (or county equivalents) nest within states – they never cross state boundaries and state boundaries always run along a set of county boundaries. Therefore, a hierarchy exists. Traditionally, only the "highest" boundary in a hierarchy is shown on Census Bureau maps. For example, although state, county, and minor civil division boundaries are coincident along a state line, only the state boundary symbol is plotted, and the map reader infers that the symbol also represents the county and minor civil division boundaries.

While this approach certainly is economical and an implied hierarchy is not a difficult concept for many users of Census Bureau maps, it may be that an explicit display of all boundaries is preferable for some groups of users. Explicitly displayed hierarchies may be quicker to interpret even for experienced users of Census Bureau maps. The drawback is that more symbols are shown along the lines where a boundary hierarchy exists and the problems with dark areas in Method 1 and space restrictions in Method 2 are compounded. More legend space for text is needed for Method 3.

There are advantages and disadvantages to each of the methods discussed above, as well as to implicit and explicit symbol hierarchies. The Geography Division at the Census Bureau, in an effort to learn the feelings of different user groups about boundary display techniques, developed a boundary interpretation and evaluation survey.

SURVEY DESIGN

A survey package was developed that included six maps of the same area with boundaries displayed in six different ways:

Map 1 - Overplotted boundaries with an explicit hierarchy
Map 2 - Overplotted boundaries with an implicit hierarchy
Map 3 - Alternating boundary symbols with an explicit hierarchy
Map 4 - Alternating boundary symbols with an implicit hierarchy
Map 5 - Unique multiple boundary symbol with key numbers and explicitly described hierarchy
Map 6 - Unique multiple boundary symbol with key numbers and implicit boundary hierarchy

The maps displayed combinations of the following boundary types: international, state, county, minor civil division, incorporated place, and American Indian reservation. The maps that used an implicit hierarchy included the following information in the legend:

> <u>IMPORTANT NOTES ON BOUNDARY INTERPRETATION</u>
>
> International boundaries are always state, county, minor civil division, and incorporated place boundaries.
>
> State boundaries are always county and minor civil division boundaries.
>
> County boundaries are always minor civil division boundaries.

In each package the maps were arranged in an order different from any other package. Respondents were asked to use the maps in the order received to complete six exercises in boundary interpretation, one exercise for each map. Once the exercises were completed, the participants were asked which methods they thought were best and worst overall, which took the least and most amount of time to interpret, which were the easiest and most difficult to interpret, and which were probably interpreted with the most and least accuracy.

Four groups of approximately thirty people each participated in the survey. Each group was chosen based on average level of experience in interpreting Census Bureau geography and maps in general:

Group 1 - Clerical/computer digitizing operators
This group was expected to have very little experience in interpreting Census Bureau geography although they had a high exposure to interpreting general map base features. This group will be involved with interpreting boundaries for certain TIGER File input operations.

Group 2 - Census Bureau professional skills development training class
This group of new (less than one year) professional employees had just completed a six week course in all aspects of Census Bureau operations, including some training in Census Bureau geography. They were expected to have some understanding of this geography and a small amount of experience with map interpretation. As Census Bureau employees they will be users of a wide array of maps.

Group 3 - Participants in the Census Bureau Boundary and Annexation Survey from randomly selected incorporated places
This group of mayors, town clerks, and city engineers was expected to have good experience with interpreting a limited scope of Census Bureau geography and a good amount of experience in map interpretation. As participants in a survey used to certify current corporate limits, they will be regular users of computer-generated Census Bureau maps.

Group 4 - Professional Census Bureau Regional Office Geographers
This group was expected to have very high levels of experience in both the interpretation of Census Bureau geography and the use of all levels of census maps. As professionals in the Census Bureau's twelve regional offices, they will be users of most computer-generated map products and will be responsible for helping many inexperienced people interpret these maps.

SURVEY RESULTS AND DISCUSSION

Following are portions of each of the six maps showing the same boundary combinations displayed in different ways. An asterisk next to the map type indicates that the notes explaining boundary hierarchy (discussed earlier) were included in the legend.

Legend for the first four maps:

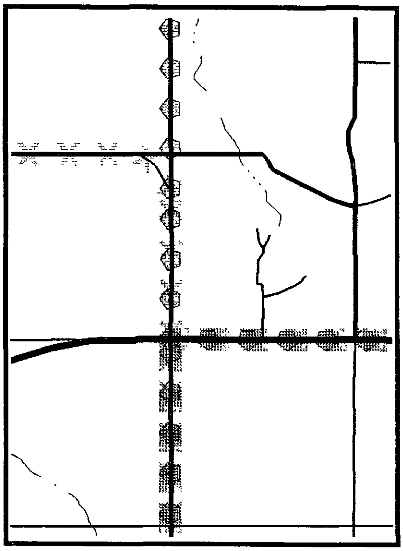
STATE

COUNTY

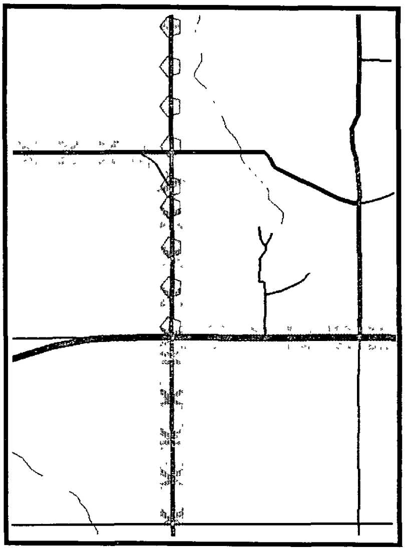
⬠ ⬠ ⬠ ⬠ ⬠ ⬠ ⬠

MINOR CIVIL DIVISION

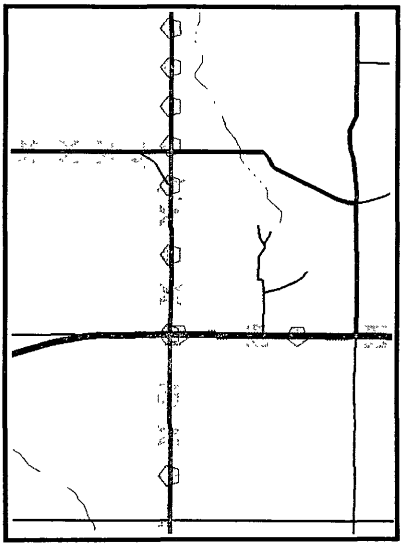
AMERICAN INDIAN RESERVATION



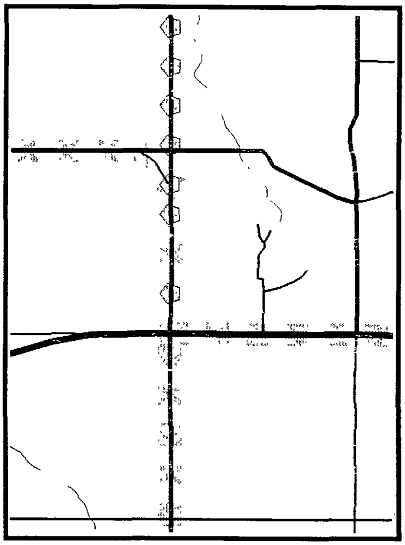
MAP 1: Overplotted Symbols, Explicit Hierarchy



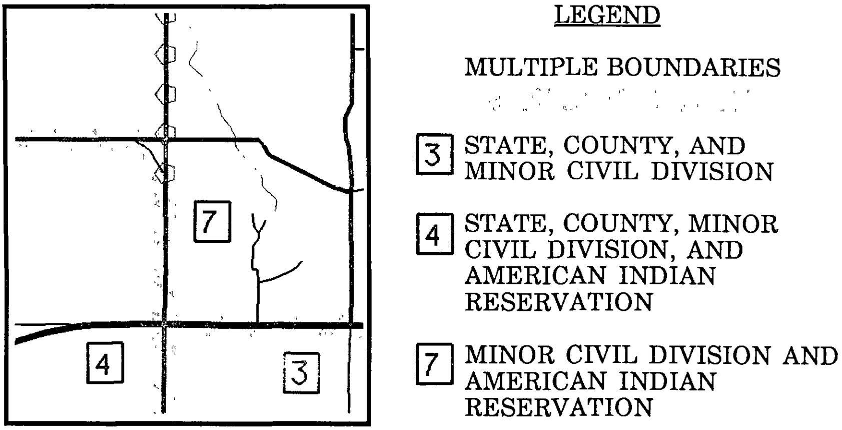
MAP 2*: Overplotted Symbols, Implicit Hierarchy



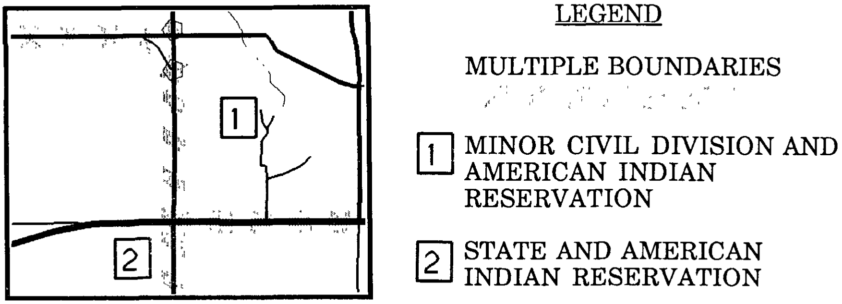
MAP 3: Alternating Symbols, Explicit Hierarchy



MAP 4*: Alternating Symbols, Implicit Hierarchy


7
4
3
LEGEND
MULTIPLE BOUNDARIES
3 STATE, COUNTY, AND MINOR CIVIL DIVISION
4 STATE, COUNTY, MINOR CIVIL DIVISION, AND AMERICAN INDIAN RESERVATION
7 MINOR CIVIL DIVISION AND AMERICAN INDIAN RESERVATION

MAP 5: Unique Multiple Boundary Symbol, Explicit Hierarchy


1
2
LEGEND
MULTIPLE BOUNDARIES
1 MINOR CIVIL DIVISION AND AMERICAN INDIAN RESERVATION
2 STATE AND AMERICAN INDIAN RESERVATION

MAP 6*: Unique Multiple Boundary Symbol, Implicit Hierarchy

TABLE 1 shows the map type (boundary display method) selected by the majority of each of the four groups in response to the criteria stated at the left. The percentage selecting each map type is indicated in parenthesis next to the map number.

TABLE 1: Map type selected in response to selected criteria, by group

CRITERIA	GROUP 1	GROUP 2	GROUP 3	GROUP 4
Boundaries are easiest to interpret	3 (43%)	3 (50%)	4 (32%)	4 (60%)
Boundaries are most difficult to interpret	1 (35%)	1 (60%)	5 (34%)	1 (79%)
Boundaries take the least time to interpret	3 (32%)	3 (54%)	3 (41%)	4 (63%)
Boundaries take the most time to interpret	1 (32%)	1 (50%)	1 (60%)	1 (63%)
Boundaries probably interpreted with the most accuracy	3 (32%)	3 (42%)	4 (38%)	4 (52%)
Boundaries probably interpreted with the least accuracy	1 (35%)	1 (55%)	1 (51%)	1 (82%)
Best overall boundary design	3 (32%)	3 (50%)	3/4 (29%)	4 (63%)
Worst overall boundary design	1 (37%)	1 (63%)	1 (51%)	1 (67%)

Map #1 is clearly the least preferred in all aspects. This is to be expected since the areas of multiple boundaries are nearly black and indecipherable on portions of the map. If we disregard that particular map, the least preferred was almost always Map 5, which used a unique multiple boundary symbol with key numbers and an explicit hierarchy.

The percentages in favor of Maps 3 and 4 are not overwhelming when considered separately; however, disregarding the method used to treat boundary hierarchy, the alternative symbol approach is clearly the most preferred method. Based on this survey the overall ranking of the six methods by map type is (in order of overall preference):

1 - Map 4 (Alternating, implicit hierarchy)
2 - Map 3 (Alternating, explicit hierarchy)
3 - Map 2 (Overplotting, implicit hierarchy)
4 - Map 6 (Unique symbol, implicit hierarchy)
5 - Map 5 (Unique symbol, explicit hierarchy)
6 - Map 1 (Overplotting, explicit hierarchy)

While the unique multiple boundary symbol did not receive high ratings, many strong comments were made by those who did favor its use. These respondents indicated that although the legend was constantly consulted, one could usually expect to retrieve the correct information regarding boundary coincidence. With regard to the alternating symbol approach, many of those preferring the implicit hierarchy stated that it took up less space than the explicit display, and the notes on hierarchy in the legend gave the necessary information for hierarchy interpretation. Those preferring the explicitly displayed hierarchy argued that users should not have to decipher a hierarchy and that it is much simpler and more consistent to have all boundaries symbolized in their positions on the map.

Regarding the ability of the participants to correctly interpret multiple boundaries, some surprising points came to light. The first is that a rather low percentage of responses were correct. The maps used in the survey had many boundaries on them because multiple boundaries were needed to satisfy the survey purpose. They were not simple maps to begin with, and in addition to the complex geography, many people are simply not accustomed to looking at maps produced on raster plotters. None of the methods used to display boundaries are familiar ones, and differently shaped screened symbols take some getting used to. These ideas may explain the low percentages of correct responses to the interpretation exercises.

TABLE 2: Percentage of each group successfully completing boundary interpretation exercise, by map number

MAP NUMBER	GROUP 1	GROUP 2	GROUP 3	GROUP 4
1	35%	37%	47%	35%
2	9%	48%	21%	58%
3	39%	52%	42%	77%
4	35%	44%	47%	46%
5	17%	48%	5%	62%
6	13%	37%	16%	48%

TABLE 2 shows that more respondents (three out of four of the groups) correctly completed exercises using Map 3 (alternating, explicit hierarchy) than any of the others. One group responded to exercises correctly most often using Map 1 (overprinting, explicit hierarchy - also that group's least favorite) and Map 4 (alternating, implied hierarchy). It was expected that users would correctly interpret the boundaries on the map they most preferred, but the results presented in TABLE 3 show otherwise. In some cases, no exercises were completed successfully by a group of respondents using their most preferred maps. While we wish to provide maps that display boundaries using a method preferred by our users, we also want them to interpret them correctly!

TABLE 3: Percentage of correct responses to exercise performed using the most preferred map

MAP NUMBER	GROUP 1	GROUP 2	GROUP 3	GROUP 4
1	0%	0%	*	*
2	20%	0%	25%	66%
3	50%	67%	50%	80%
4	0%	50%	25%	42%
5	*	50%	0%	50%
6	40%	25%	*	50%

*Not selected as the best overall map by anyone in this group.

In summary, the findings of this survey are:

1) Alternating boundary symbols were the most preferred, with implicit hierarchies slightly preferred over explicit hierarchies
2) Alternating boundary symbols using an explicit hierarchy were correctly interpreted more often than those using an implicit hierarchy.

Maps produced by the Census Bureau on electrostatic raster plotters will be very different from those produced in the past. Although current technology allows us to accomplish in a quick and efficient manner tasks that in the past were cumbersome, some concessions must be made. The advantages of speed and flexibility of electrostatic plotters outweigh the fact that the output is not a highly-polished product. Although not publication quality, the output is entirely sufficient for Census Bureau field operations. Adjustments must be made by Census Bureau employees and outside users alike in learning to use the new products. By including our map users in our map design process, we hope to make the move from traditional, manually drawn maps to computer-generated maps an easier one.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the work of Amy Bishton of Mapping Operations Branch, Geography Division, U.S. Bureau of the Census, who wrote the programs that generated the maps used in this survey, and Lee Ebinger, Mapping Operations Branch, who programmed all of the boundary symbology routines.

AUTOMATED MAP INSET DETERMINATION

Frederick Roland Broome
Constance Beard
April A. Martinez
Geography Division
U.S. Bureau of the Census
Washington, D.C. 20233

ABSTRACT

Cartographers typically use their judgement to determine the need for an inset by visually inspecting a map and subjectively identifying the areas which are too dense for adequate feature or text placement. To determine insets an automated system must emulate the human decision process. This paper considers eight different approaches for automating map inset determination. It describes the development of a system for Census Bureau needs that is based on the most efficient of the eight methods.

INTRODUCTION

There are many situations where a map design must provide adequate space for text placement within areas delineated by map features. This is particularly true for most census field operations maps. These maps require space for placing census block numbers and sometimes space for marking residential structure locations along a street. Since maps used for census field operations are constrained by sheet size and number of sheets, the sizes of the areas for text placement on portions of the maps can become too small and insets must be produced.

Standard, good cartographic procedure calls for each map to be designed, examined, and on sheets where the text cannot be adequately placed, either the base map is enlarged or insets prepared . The field maps for the 1990 Decennial Census of Population and Housing of the United States will be prepared by computer programs run in a non-interactive, batch mode. Hundreds of thousands of maps, each different must be produced within a few months. There will be no opportunity to prepare "trial" maps either as hard copy plots or on a graphics terminal screen. If cartographic principles of map design are to be applied, they must become part of the computer programs used to generate the map plots. The map volume and short production schedule necessitates an efficient method of identifying when and where an inset is needed. This paper describes the research, development and implementation of an automated inset determination procedure.

DISCUSSION

Cartographers typically use their judgement to determine the need for an inset. They visually inspect the map and subjectively identify the limits of areas which are "too crowded", "too small", or "too dense." It was apparent that to determine insets an automated system must emulate the human decision process. The human process appears to arrive at a decision by answering the following three questions.

What features are considered when determining density?
What constitutes "too dense?"
What are the bounds of dense areas to determine inset limits?

Operationally there are two ways to execute automated inset determination: by preprocessing the files and storing the information or at the time of production. Each method has advantages and disadvantages. Extraction at time of map production means that the effort is expended for one scale (the scale of the map being produced) and that the extraction must be done every time the map is generated. This method has lower requirements for storing and updating of control information than the extraction in advance method. Extraction in advance allows for tailoring the system for use at many scales is only redone when the TIGER File partition from which the extract has been made is modified.

Since the time it takes the computer to determine the insets depends upon the efficiency of the algorithms, various algorithms were developed and tested. The algorithms developed were specific to the Census Bureau's TIGER File structure. A different file structure will probably yield different efficiencies, but the procedures for inset determination will be similar.

The TIGER File structure is based upon entities known as zero-cells, one-cells and two-cells. A full description of these entities and related files is provided in other papers presented at this conference*. For purposes of discussion, the zero-cells can be considered as the intersection points of features and, therefore, the endpoints of one-cells. One-cells are the lines connecting zero-cells and bounding two-cells. All the other points along the feature between the endpoints are stored in an auxiliary file. The two-cells are the smallest areas bounded by one-cells. Aggregates of two-cells make up higher level geography such as census blocks.

Eight different algorithms were proposed and tried. The eight approaches tested all aspects of the spatial data available from the TIGER file. Several other methods were proposed and immediately discarded as computationally too expensive. For example, computing the average length of all the one-cells necessitates use of the distance formula.

The underlying principle is based upon feature counts within a grid cell. A grid with a known cell size is determined for an area to be mapped. The TIGER File is then read and the number of occurrences for each item within a cell is calculated. The resulting matrix of values is then smoothed by summing the counts for groups of nine cells and recording the results as the value for the center cell of the 3 X 3 group. The smoothing operation removes local irregularity due to the use of a single coordinate to represent a linear and/or areal feature.

*See papers by Kinnear, Knott, and Meixler in these proceedings for discussions of the TIGER System.

Algorithms Selected for Consideration

The following algorithms were selected for testing:

1. One-cell, midpoint method. The coordinates of the endpoints of the one-cells are added together and divided by two to get a midpoint.

2. One-cell, average of all points method. The coordinates of all the points along the one-cell and the endpoints are added and the results divided by the count to get an average.

3. One-cell, endpoint method. The zero-cells are used.

4. Two-cell, envelope midpoint method. The maximum and minimum coordinates of the two-cells are added and divided by two to get a midpoint.

5. Two-cell, weighted area centroid method. The area and geographic centroid of the two-cell is determined and the value at the centroid is the area.

6. Census block, envelope midpoint method. The maximum and minimum coordinates for each census block is determined by aggregating the two-cells which constitute the block. The sum is divided by two and the resulting coordinates are used.

7. Census block, two-cell average centroid method. The two-cell average centroid is derived by adding all the maximum and minimum coordinates of the two-cells and dividing by the count to get an average.

8. Census block, weighted area centroid method. The area and geographic centroid of the block is determined and the value at the centroid is the area.

A program was developed for each method and run using a TIGER 87 file partition from the 1987 Test Census. These programs extracted the features from the TIGER File, computed the centroid in latitude/longitude (and the area if required), and stored the results in intermediate files. The points calculated by these programs were analyzed for determination of dense areas. The two methods based upon area calculation were discarded immediately as computationally inefficient when their dot patterns and computer times were compared to the other methods.

Census field operations are primarily concerned with roads as access to the population and as statistical boundaries. For this, our concern is to identify dense areas of road features. The one-cell, endpoint method was discarded because of the difficulties in avoiding multiple counting of the endpoints.

The census block, two-cell average centroid method was discarded because most blocks consist of one two-cell. Where blocks consist of two or more, two-cells, the difference of the centroid from the whole block midpoint position is insignificant. Thus, the extra computation was deemed inefficient.

The four methods retained for the second developmental phase were: the one-cell midpoint; the one-cell, average of all points; the two-cell envelope midpoint and the census block envelope midpoint. The intermediate files for the four methods were processed through a program which produced a grid cell count matrix, i.e. the number of feature centroids that fell within each grid cell. Each grid cell was .25 inches on a side. The number of grid cells along each axis is determined by converting the differences between the minimum and maximum longitude and latitude into inches at map scale and then dividing by the grid cell side size. The map can never be greater than 36 x 36" because of plotter paper size and map design constraints. Consequently the grid never was limited to no more then 144 cells to a side at the selected cell size. The four count matrices were printed and their patterns analyzed.

After the second phase of development, the one-cell average of all points was discarded because there were no apparent differences between the results of this method and those of the one-cell, midpoint method. Then the census block envelope midpoint method was dropped because of the many computations and file accesses required to determine the block envelope. Also the fact that text placement must avoid conflict even with the other two-cell boundary features within the block. Together, these made this method inefficient when compared to the remaining methods.

The one-cell, midpoint is computationally the easiest of the linear feature methods. The two-cell, midpoint method is the easiest of the areal methods, and it is particularly easy when using the TIGER File since the centroid coordinates of the two-cell are stored as part of the base file information. The execution time for both methods is directly proportional to the number of features processed. Thus, the areal method is faster. The execution times are also low because the programs are accessing elementary units of the TIGER File and only performing additions when required, a binary shift for division by two.

Calibrating the Inset Determination Algorithm

The next phase in the development was to introduce human cartographic expertise. Ten professional cartographers, with an average experience of five years in census map making were asked to examine the same maps that the computer programs were producing and to mark on overlays the extent of the insets needed, if any. Their only guidance was that the maps were to show census block numbers. The cartographers marked the overlays without discussion among themselves and without knowledge of what the previous cartographers had marked.

The results showed a surprisingly close match between the cartographers. The variances averaged less than one-fourth inch at map scale. This variance was within the grid cell size and considered quite good for the intended purpose.

Next a smoothed count matrix was plotted out at the scale of the map used by the cartographers. The overlays were placed over the plotted count matrix. A visual examination revealed that the human cartographers placed their inset boundaries so that they enclosed grid cell clusters with counts above about one-half the difference between the maximum and minimum cell counts. This not only worked for large dense areas, but it identified smaller dense areas. Significantly timeframe, it avoided picking up extremely small clusters of one or two census blocks where an inset would be inappropriate.

The numeric value that was developed from the cartographer's efforts to delimit insets was shown to be a function of grid cell size, map scale, and size of text to be plotted. The grid cell size and size of the text to be plotted remained the same throughout development. Only the map scale varied.

Final System Description

The automated map inset determination system developed is based upon the two-cell envelope midpoint method and uses the human expert derived factors. It also processes a map at a time for insets rather than preprocessing a whole TIGER File partition. The preprocessing approach may be reevaluated when TIGER Files become available in sufficient quantity to compare the results. The two-cell envelope midpoint method was selected because the major problem in current field map design is census block number text placement.

The system operates in the following steps:

1. For a whole TIGER File partition, compute the latitude and longitude midpoint of all two-cells that are bounded by at least one one-cell that is a road and store the results in an intermediate file.
2. For all the two-cells within a given map sheet image area convert the latitude and longitude midpoints from the intermediate file into inches at map scale and store these in a temporary file.
3. Process the temporary file through the program that creates the grid cell count matrix and smooths it. Then use the smoothed matrix to determines the need for and limits of insets. For each inset determined, a record is put out to an inset control file. The control file record contains the inset latitude/longitude windows for that map.

Future Research

The current system was developed within a short timeframe imposed by production schedules. While the system performs the task for which it was designed, many questions remain. Will the expert derived numeric value continue to be adequate when the TIGER Files for more geographic areas are available? Is the relationship between map scale, size of text to be plotted, grid cell size and the numeric value linear or does some other relationship exist? Is there a more efficient programming sequence? What is the effect of other map features on the need for an inset and how is it detected? These and other questions are currently being researched, but many areas for study remain.

ASSESSING COMMUNITY VULNERABILITY TO HAZARDOUS MATERIALS WITH A GEOGRAPHIC INFORMATION SYSTEM

Robert B. McMaster
James H. Johnson, Jr.
Department of Geography
University of California, Los Angeles
Los Angeles, California 90024

ABSTRACT

The purpose of this study is to demonstrate the utility of a risk assessment model as an anticipatory hazardous management tool using a grid-based geographic information system. Specifically, the risks resulting from the on-site storage of hazardous materials and from transport of dangerous commodities through a city has been analyzed. Santa Monica, California, was one of the first cities in the U.S. to enact a hazardous materials disclosure ordinance and, therefore, was selected for the community vulnerability analysis. A comprehensive geographic data base of Santa Monica was developed at a 100 meter resolution. In all, fifty variables were incorporated into the data base, including transportation networks in varying detail, traffic volume along major routes, ethnicity, population density, age structures, landuse, earthquake fault lines, institutions, elevation, and nine categories of hazardous material based on the United Nations classification. These data were then analyzed using the Map Analysis Package (MAPS) developed at Yale University in order to derive a series of maps depicting the contoured risk surface of the city. Based on the results of this analysis, a set of strategies designed to reduce the risk of hazardous materials incidents in Santa Monica are being formulated in conjunction with local emergency managers.

INTRODUCTION

In the United States incidents involving unintentional releases of hazardous materials into the environment occur frequently. A recent EPA study revealed, for example, that during the first five years of this decade about five accidents a day resulted in the release of toxic materials into the environment from small and large production facilities (Diamond, 1985). During the ten year period ending in 1983, there were 126,086 transportation accidents involving accidental releases of hazardous materials, an average of nearly 13,000 a year. These incidents claimed 260 lives, caused more than 700 injuries, and resulted in property and equipment damage in excess of $146 million (U.S. Department of Transportation, 1983). In addition, during the ten year period ending in 1982, there were 18,470 gas pipeline failures which claimed 340 lives and injured another 3536 people. Nearly two-thirds of these pipeline failures were attributed to damage, caused by excavations, and the remainder to corrosion, construction defects, and material failures (U.S. Department of Transportation, 1984).

The U.S. government has taken steps both to reduce substantially the occurrence of hazardous materials incidents and to minimize the potentially adverse effects on people and property when accidents actually occur (Hohenesmer, Kates and Slovic, 1983). It has been estimated, for example, that in 1979 the U.S. spent $30 billion on hazard mitigation and emergency preparedness (Hohenesmer and Kasperson, 1982). However, we tend to concur with Tierney when she states that "local emergency personnel are in the best position to know about the hazards in their own community" (1980, p. 78). Building upon this view, Johnson and Zeigler (1986) have developed a simple risk assessment model which should enable local emergency managers to determine the extent to which their communities are vulnerable to hazardous materials incidents. Their risk assessment model requires local emergency managers first, to identify the hazards present in their community and to map the hazard zone each enscribes on the landscape; second, to superimpose on the map population distribution and land-use data; and third, to use this information to develop site-specific strategies to reduce the risks and to mitigate the potential negative consequences should an accident occur. Application of the model requires local jurisdictions to enact legislation which stipulates, as part of the licensing process, that businesses disclose the kinds and amounts of hazardous materials used, generated, or stored onsite. These data serve as the basis for the development of a comprehensive hazardous materials tracking system which, in turn, is used to assess the vulnerability of population and areas within local jurisdictions to hazardous materials incidents.

In this paper, we focus on application of the hazardous materials risk assessment model developed by Johnson and Zeigler (1986), advocating the use of a geographic information system called MAP (Map Analysis Package) in the hazards identification and community vulnerability assessment process. Toward this end, we shall proceed in the following manner. First, we outline the major components of MAP and discuss the requisite data bases. Next, we apply MAP in the actual hazardous materials risk assessment, arriving at a series of maps depicting the contoured risk surface of the City of Santa Monica, one of the first municipalities to enact a hazardous materials disclosure ordinance (Staff Reporter, 1985). Finally, we identify several steps which can be taken to reduce substantially the risks of hazardous materials incidents in Santa Monica.

MAP AND THE DATABASE

Since it was decided to geocode data in raster format, the Map Analysis Package (MAP) developed at

Yale University was selected for storing and analyzing the spatial data. The four significant capabilities of this package,including reclassification, overlay, cartographic mensuration, and neighborhood analysis, proved ideal in revealing the relationship between the hazardous materials and the population and institutions at risk. Currently, a vector-based system is being developed in order to generate contour and 3-D mapping capability.

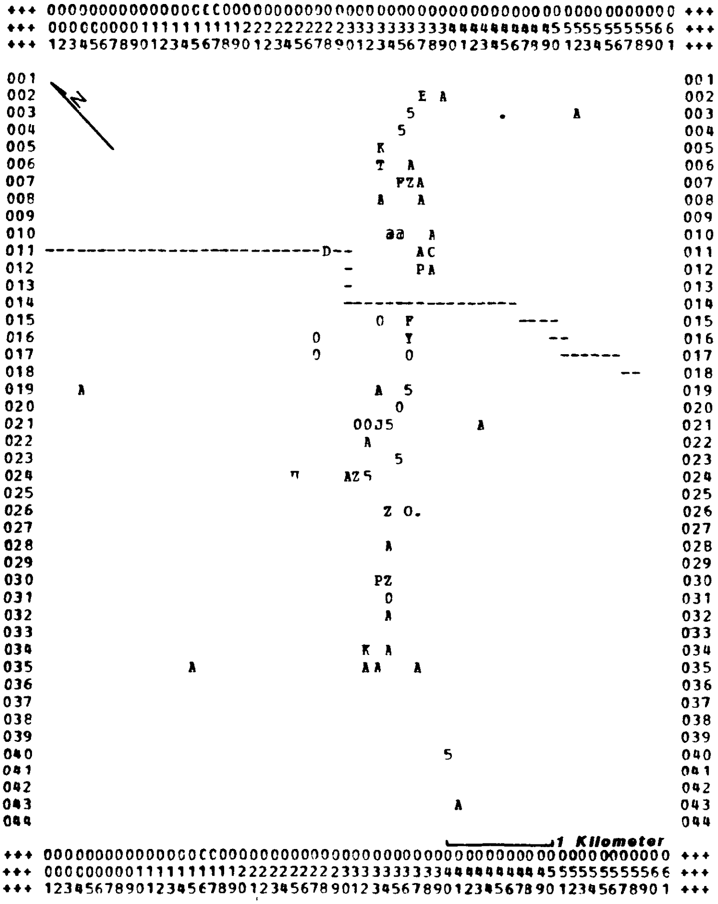
For the purposes of this study, four categories of data (demographics, landuse, physiography, and hazardous materials) were geocoded at a resolution of 100M and entered into the Map Analysis Package (MAP) (Table 1). The demographic data were taken from the 1980 Census of Population and Housing. The land use data were obtained from the Santa Monica City Planning Office. The City's Office of Emergency Preparedness provided the data pertaining to the types and location of hazardous materials.

Category	Specific variables
Demographic	population under 5, 5 to 15, 15 to 65, and over 65, language, population density, percent white, asian, black, hispanic, other minority, institutions
Land Use	land use, storm drains, transportation, freeway, roads, traffic flow
Physiography	topography, earthquakes
Hazardous Materials	explosives, gases, flammable liquids, flammable solids, oxidizers and organic peroxides, poisonous and infectious materials, radioactive materials, corrosives, miscellaneous hazardous materials, cleaners, gunshops, major oil pipes, polychlorinated Biphenyls (PCBs) and underground gas tanks

Table 1. Variables Included in the Data Base

ANALYSIS

Figure 1 depicts the distribution of hazardous materials in Santa Monica. In developing this map, we utilized the United Nations Classification of Hazardous Materials as well as Zeigler, Johnson, and Brunn's (1983) typology of technological hazards to arrive at a total of fourteen classes of hazardous materials. As the figure shows, many of the sites were found to contain only 1 or 2 types of hazardous materials, but others contained as many as five categories. For example, two sites, located in close proximity to the Santa Monica Freeway, stored flammable liquids, explosives, gases, poisons, and corrosives. In general, the



Types of Hazardous Materials

----------	1	OILPIPE			
....	5	CORROSIVE (CR)			
0000000000	10	RADIOACTIVE (RA)	UUUUUUUUUU	40	FL; RA
5555555555	15	POISONOUS (PS)	ZZZZZZZZZZ	45	FL; PS
AAAAAAAAAA	20	PS; CR	0000000000	50	FL; PS; CR
DDDDDDDDDD	23	OILPIPE; PS; CR	5555555555	55	FL; PS; RA; CR
FFFFFFFFFF	25	PCBS(PB); PS; CR	EEEEEEEEEE	60	OXIDZR(OX);FL;CR
KKKKKKKKKK	30	FLM LIQ(FL) PS;CR	JJJJJJJJJJ	65	FLM SOLID(FS);FL
PPPPPPPPPP	35	FLM LIQUD(FL);CR	0000C00000	70	GAS(S) (GS); FL
			TTTTTTTTTT	75	FL; GS; PS
			YYYYYYYYYY	80	EXPLSVE(EX);FLPS
			@@@@@@@@@@	85	FL;EX;GS;PS;CR

Figure 1. Types of Hazardous Materials in Santa Monica

distribution of hazardous materials parallels the Santa Monica Freeway in a northeast-southwest direction.

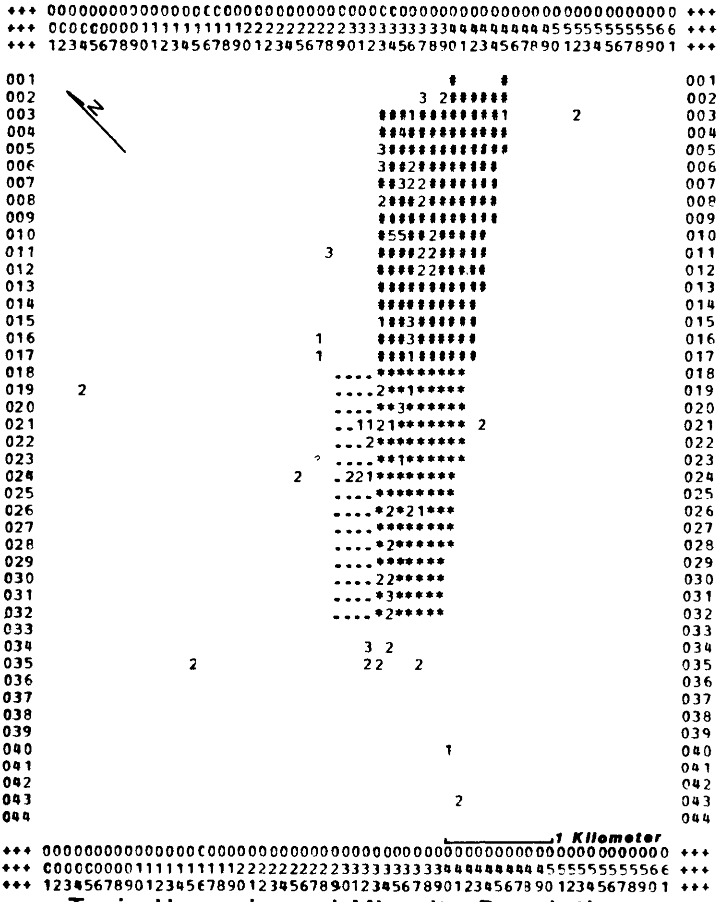
Following the methodology proposed by Johnson and Zeigler (1986), the next step in the risk assessment process was to identify both the population and the institutions at risk to hazardous materials in Santa Monica. For the purpose of determining the population at risk, the actual number of hazards per grid cell was used instead of the types of hazards in each cell, as in Figure 1. Analyses of the demographic data revealed that three subgroup of the population are especially vulnerable to hazardous materials. Minority group members in Santa Monica (i.e., Asians, Blacks, and Hispanics) reside extremely close to the "high intensity" storage of hazardous materials--basically paralleling the Santa Monica Freeway (Figure 2). Directly adjacent to this "hazardous corridor," there exists a high concentration of population under age 5 and over age 65 (Figure 3). For reasons discussed in detail elsewhere (Perry 1985), all three of these groups may very well require special attention and assistance should an incident involving the unintentional release of hazardous materials into the local environment occur.

The distribution of institutions in Santa Monica is depicted in Figure 4. To determine which of these institutions are at risk, the spatial operators existing with MAP were used to create an "at risk" buffer zone should a hazardous material incident occur. For the purpose of illustration here, we assumed that an area within 500 m of the hazard materials site would be at risk. We then superimposed on this map seven categories of institutions which can be found in Santa Monica (Figure 4). As Figure 5 shows, many of the institutions lie within the "at risk" zone as it is defined for the purpose of this analysis. Of particular note is the significant number of schools (S) and hospitals (H) which lie immediately adjacent to sites which either produce, store, or use hazardous materials.

CONCLUSION

In this paper we have attempted to demonstrate the utility of the Map Analysis Package in the identification of community vulnerability. Our analysis revealed that in Santa Monica both the resident population (mainly Blacks, Hispanics, and Asians) and the institutions (especially schools and hospitals) along the freeway are most vulnerable to the risks of hazardous materials incidents. Based on these results, local emergency planners can take several steps to (1) minimize the probability of such incidents and (2) reduce substantially the risks to public health and safety in the event of such an accident. These include:

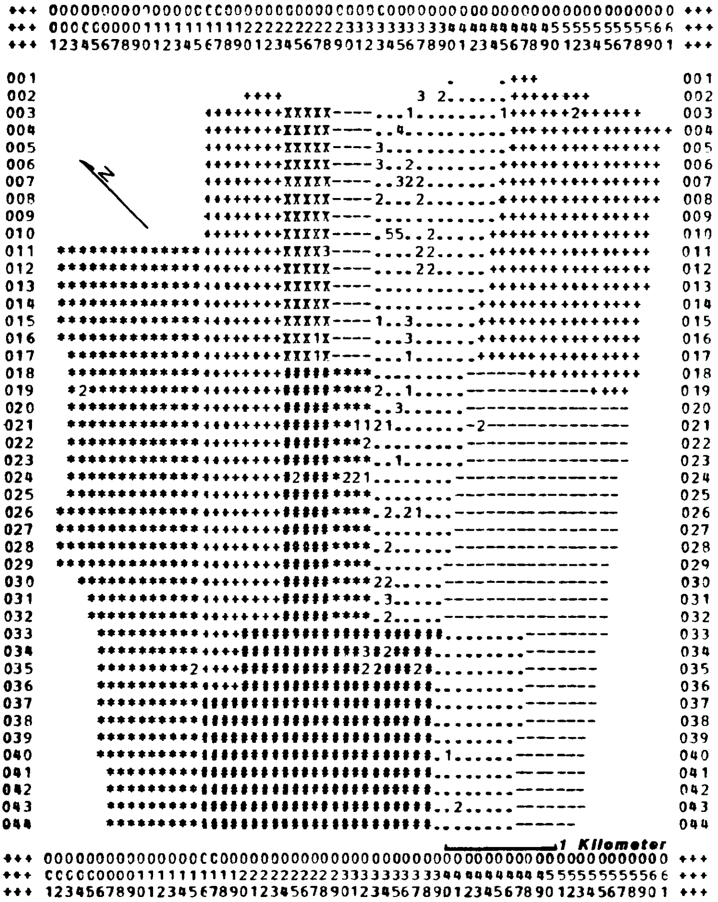
(1) Conduct periodic inspection of local businesses to insure that hazardous materials are being handled safely.



Toxic Hazards and Minority Population

```
HAZMINTY


                 0   ................     2306 CELLS   85.9%

1111111111       1   1 TOXIC HAZARD         14 CELLS    0.5%

2222222222       2   2 TOXIC HAZARDS        32 CELLS    1.2%

3333333333       3   3 TOXIC HAZARDS        10 CELLS    0.4%

4444444444       4   4 TOXIC HAZARDS         1 CELLS    0.0%

5555555555       5   5 TOXIC HAZARDS         2 CELLS    0.1%

..........      31   10-15% BLACK POP       54 CELLS    2.0%

**********      32   24% BL 14% HS         108 CELLS    4.0%

##########      33   8%AS 24%BL 21%HS      157 CELLS    5.8%
```

Figure 2. Toxic Hazards and Minority Population



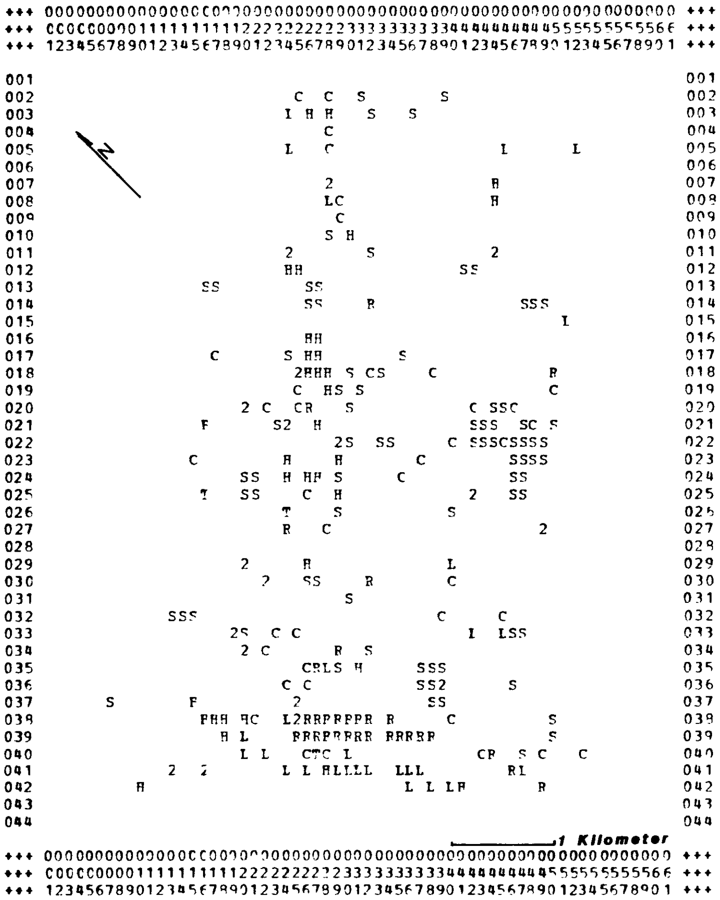
Toxic Hazards and Population Under 5/Over 65

HAZPOP

1111111111	1	1 HAZARD
2222222222	2	2 HAZARDS
3333333333	3	3 HAZARDS
4444444444	4	4 HAZARDS
5555555555	5	5 HAZARDS

	7	UND 5	OVER 65
....	8	1-2%	5-9%
....	9	8-9%	5-9%
----------	12	3-4%	10-14%
----------	13	5%	10-14%
++++++++++	16	1-2%	15-19%
++++++++++	17	3-4%	15-19%
**********	18	5%	15-19%
**********	21	1-2%	20-24%
XXXXXXXXXX	22	3-4%	20-24%
##########	26	1-2%	25-35%

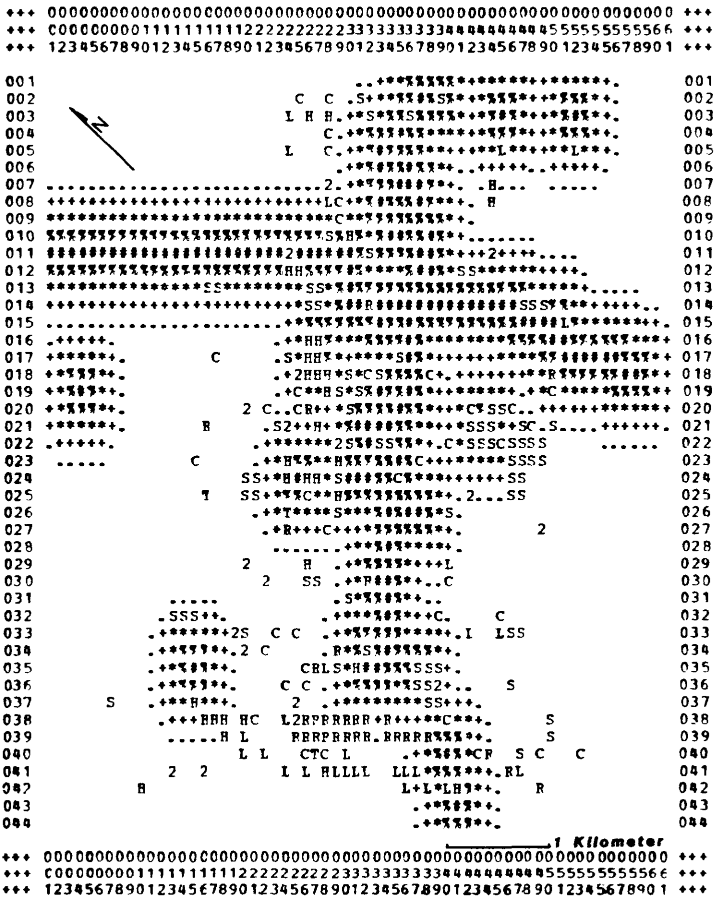
Fifure 3. Toxic Hazards and Population Under 5 / Over 65



Services

```
SERVICES

                    0  ................   2451 CELLS   91.3%

CCCCCCCCCC         21  CHURCH(S)            39 CELLS    1.5%

RRRRRRRRRR         22  COMM SERVICE(S)      32 CELLS    1.2%

TTTTTTTTTT         23  MOVIE THEATRE(S)      3 CELLS    0.1%

HHHHHHHHHH         24  HOSPITAL(S)          33 CELLS    1.2%

SSSSSSSSSS         25  SCHOOL(S)            80 CELLS    3.0%

LLLLLLLLLL         26  LODGING              28 CELLS    1.0%

2222222222         27  > THN 1 OF ABOVE     18 CELLS    0.7%
```

Figure 4. Institutions in Santa Monica



Services and Hazardous Buffer Zone

	0	0.5 KM TO HAZARD	1234 CELLS	46.0%
....	1	0.4 KM TO HAZARD	207 CELLS	7.7%
++++++++++	2	0.3 KM TO HAZARD	260 CELLS	9.7%
**********	3	0.2 KM TO HAZARD	318 CELLS	11.8%
%%%%%%%%%%	4	0.1 KM TO HAZARD	320 CELLS	11.9%
##########	5	TOXIC HAZARD	112 CELLS	4.2%
CCCCCCCCCC	21	CHURCH(S)	39 CELLS	1.5%
RRRRRRRRRR	22	COMM SERVICE(S)	32 CELLS	1.2%
TTTTTTTTTT	23	MOVIE THEATRE(S)	3 CELLS	0.1%
HHHHHHHHHH	24	HOSPITAL(S)	33 CELLS	1.2%
SSSSSSSSSS	25	SCHOOL(S)	80 CELLS	3.0%
LLLLLLLLLL	26	LODGING	28 CELLS	1.0%
2222222222	27	> OF ABOVE	18 CELLS	0.7%

Figure 5. Institutions (SERVICES) and Hazardous Buffer Zone

(2) Design public education programs which inform the population at risk of not only the potential hazards that exist in their vicinity, but also of the range of protective actions possible in the event of an accident.

(3) Develop emergency response plans to evacuate both the resident and institutional population from the "high risk" corridor along the freeway.

(4) Identify host facilities which could serve as emergency relocation centers for both the resident and institutional population.

ACKNOWLEDGEMENTS

The authors wish to thank Robine Lewis for her computer assistance on the project and the UCLA Academic Senate for financial support.

REFERENCES

Diamond, S. "U.S. Toxic Mishaps in Chemicals Put at 6928 in Five Years," *New York Times*, October 3, 1985, p. 3.

Hohenesmer, C. and J.X. Kasperson (eds) *Risk in Technological Society*, Boulder, Colorado: Westview Press, 1982.

Hohenesmer, C., R. Kates, and P. Slovic, "The Nature of Technological Hazard," *Science*, April 22, 1983, pp. 378-384.

Johnson, J.H.,Jr. and D.J. Zeigler, "Evacuation Planning for Technological Hazards: An Emerging Imperative," *Cities*, Vol, 3, 1986, pp. 148-156.

Perry, R.W. *Comprehensive Emergency Management: Evacuating Threatened Populations*. Greenwich, CT: JAI Press, 1985.

Tierney, K.J. *A Primer for Preparedness for Acute Chemical Emergencies*, Disaster Research Center, Ohio State University, 1980.

U.S. Department of Transportation, *Annual Report on Hazardous Materials Transportation 1983*, Washington, D.C., 1984.

U.S. Department of Transportation, *Annual Report on Pipeline Safety Calendar Year 1982*, Washington, D.C., 1984.

Zeigler, D.J., J.H. Johnson, Jr. and S.D. Brunn, *Technological Hazards*. Resource Publications in Geography. Washington, D.C.: Association of American Geographers.

IMPROVEMENT OF GBF/DIME FILE COORDINATES IN A GEOBASED INFORMATION SYSTEM BY VARIOUS TRANSFORMATION METHODS AND "RUBBERSHEETING" BASED ON TRIANGULATION

Dr. George L. Fagan and Henry F. Soehngen
Bowne Management Systems Inc.
235 East Jericho Turnpike
P.O. Box 109
Mineola, New York 11501
(516) 248-6840

ABSTRACT

The Geobased Information System for the Town of Oyster Bay, Long Island, NY stores a street network represented by a GBF/DIME file and tax parcel centroids. When layers of coordinate information from the various sources are displayed on a screen or drawn on maps, obvious misfits of parcel centroids within logical blocks delineated by GBF/DIME segments have been encountered, leading to the need to "clean-up" the views for aesthetic and planning purposes.

The large amount of coordinate data for the 114 square mile area with over 96,000 parcels, made automation of "correcting" the coordinates to a common base essential. It was decided to use the Parcel Centroid layer as the base and transform the GBF/DIME coordinates to this base. Control points were established on the Parcel Centroid layer as apparent street intersections and digitized. The corresponding GBF/DIME coordinates were then transformed by least squares using a number of transformation procedures including linear affine, bilinear affine, orthogonal, and 2nd order polynomial.

The results while satisfactory in some cases, still left residuals of the control points which could only be resolved by arbitrary rules. Therefore, for the final production, a transformation method based on triangulation of the control point network by Delaunay triangles followed by use of linear affine transformation parameters for each triangle was employed. The results were highly satisfactory.

INTRODUCTION

The Town of Oyster Bay Geobased Information System (GIS) serves to provide highly useful graphical and non-graphical data for planning, including pavement management and the routing of solid waste collection vehicles. The GIS stores a street network represented by a GBF/DIME file, parcel centroids and associated files containing other features and information.

Created during a project to develop computerized collection routes for the Town's solid waste collection vehicles (Fagan 1986), the Geobased Information System can accomodate municipal information such as maps, assessment records, census information, and land use information as though it were a single continuous map. The database is extremely flexible in its design and permits Town departments to access geographic and related attribute information from a single source. This centralization provides efficiency from reduced record maintenance and the availability of consistent information.

Database Development

The Geobased Information System presently consists of the following base map and overlay features:

- Street Network
- Non-Street Features
- Municipal Boundaries
- Parcel Centroids
- Pavement Management Information
- Solid Waste Collection Information
- Annotation

Various data sources were employed to gather the above information and create the GIS. Several of these sources were available on computer media and included the GBF/DIME file for the Nassau-Suffolk County Standard Metropolitan Statistical Area (SMSA) from the U.S. Bureau of the Census, the Nassau County Parcel Coordinate/Area File, and the Town of Oyster Bay Assessment File.

The street segments from the GBF/DIME file were used to define a graphic network of all streets within the Town and a non-graphic database of attribute information for each segment. This network was checked for missing or incorrect streets, improper street geometry and topology and incorrect street names or other attribute information. The database was updated to reflect any necessary changes.

The Town's Assessment File contained useful information but lacked coordinate information which prevented it from being integrated into the GIS. Coordinate information was available for each parcel in Nassau County's Parcel Coordinate/-Area File. A computer program was written to merge these two files and bulk load the information as the Parcel Centroid layer within the GIS.

Misfit of the GIS Overlays

When the layers of coordinate information from the various sources used to create the GIS were displayed on a screen or drawn on maps, obvious misfits of parcel centroids within the logical blocks delineated by GBF/DIME segments were encountered (see Figure 1). These positional discrepancies, which often result when map data from different sources are combined or overlaid onto a single map, are usually the result of differing source, projection and accuracy requirements of the original documents.

These misfits lead to the need to "clean-up" the views for aesthetic and planning purposes. The large amount of coordinate data for an area of 114 square miles and over 96,000 parcels made automation of "correcting" the coordinates to a common base very essential.

Establishment of Control Points

In order to improve the positional relation between map layers in the Geobased Information System, it was decided to use the Parcel Centroid layer as the base and transform the GBF/DIME coordinates to this base. To accomplish such a transformation, definition of control points was necessary. These control points are used to force registration of selected GBF/DIME intersections over their apparent counterparts in the Parcel Centroid layer and in turn bring other intersections into near coincidence with their counterparts.

The transformation routines, which are performed interactively at a graphics workstation, were implemented so as to allow an iterative, piecewise solution (White and Griffen 1985) over successive user defined polygons until a suitable solution is obtained. These routines include a utility that assists in the insertion of control points.

The interactive routine to place control points allows the user to select an apparent intersection location in the Parcel Centroid layer, digitize a control point at this location, and assign the corresponding GBF/DIME intersection (see Figure 1). All required attribute information regarding the control points is automatically collected from the GIS and the control points are stored in a data layer created for this purpose.

The rules used for control point placement were subjective and based primarily on the orientation of misfit between the layers for the defined polygon. General guidelines included:

- Include control points near the corners of the polygon.
- Place control points strategically around the periphery of the polygon.
- Place several control points near the center of the polygon.
- Do not place control points too close together.
- Control should be placed in locations of extreme offset.
- For more precise coincidence, add more control points. A minimum of 8 control points was recommended per polygon.

EXPERIMENTATION WITH GLOBAL MAP TRANSFORMATIONS

The literature in photogrammetry and remote sensing abounds with the use of various mathematical transformation functions to eliminate systematic and random errors in observational data to improve results (Kratky 1972, Bahr 1976). Experience with these methods seemed to indicate that one or more of the standard transformation concepts would produce satisfactory improvement in the final composite map product and database.

Transformation Models Utilized

2 Dimensional Affine Transformation. [6 Parameters]

$$X = a_1 + a_2x + a_3y \quad (1)$$

$$Y = b_1 + b_2x + b_3y \quad (2)$$

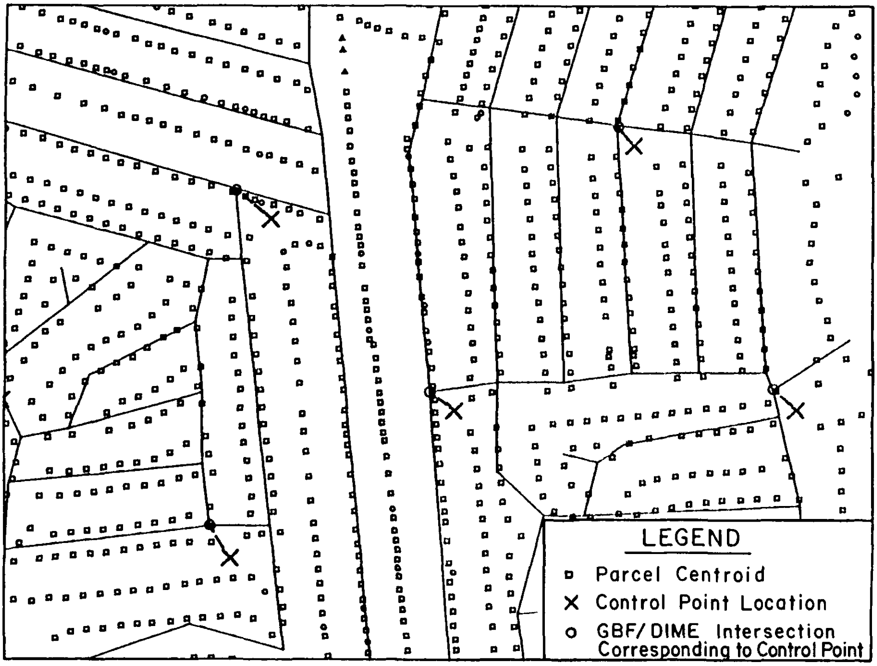


FIGURE 1
Misfit of Parcel Centroids within Logical Blocks Delineated by GBF/DIME Segments

Bilinear Polynomial. [8 Parameters]

$$X = a_1x + a_2y + a_3xy + a_4 \quad (3)$$
$$Y = b_1x + b_2y + b_3xy + b_4 \quad (4)$$

Linear Conformal (Similarity Transformation). [4 Parameters]

$$X = a_1x - a_2y + a_3 \quad (5)$$
$$Y = a_1y + a_2x + a_4 \quad (6)$$

2nd Order Polynomial. [12 Parameters]

$$X = a_1 + a_2x + a_3y + a_4x^2 + a_5y^2 + a_6xy \quad (7)$$
$$Y = b_1 + b_2x + b_3y + b_4x^2 + b_5y^2 + b_6xy \quad (8)$$

Projective Transformation. [8 Parameters]

$$X = (a_1x + b_1y + c_1)/(d_1x + e_1y + 1) \quad (9)$$
$$Y = (f_1x + g_1y + h_1)/(d_1x + e_1y + 1) \quad (10)$$

In the above equations:
- X,Y are the control point coordinates in the parcel centroid layer.
- x,y are the corresponding original GBF/DIME coordinates for the control points.
- $a_1, a_2, \ldots, a_6$; $b_1, b_2, \ldots, b_6$; and $a_1, b_1, c_1, \ldots, h_1$ are required transformation parameters.

Because the number of available control points per map area always exceeded the minimum number required to obtain the transformation parameters for a particular transformation model (by design), a Least Squares solution for the determination of the particular parameters was always utilized.

Least Squares Principles

A solution based on a unit matrix to represent the weights assigned to the control points was considered to be most logical. Thus, the system of observation equations for the Least Squares method can be expressed in matrix form as:

$$AX - L = V \quad (11)$$

where:

X is the vector of required transformation parameters.
A is the coefficient matrix of observation equations.
L is the vector of observed quantities.
V is the vector of residuals.

Then,

$$A^T AX = A^T L \quad (12)$$

is a system of normal equations (number of equations equals number of unknowns) which can be solved for the unknown parameters of the particular transformation model chosen.

The solution for the elements of the X vector can be obtained by a number of elimination methods or by matrix inversion operations. In the computer program system to be described, the solution vector elements were obtained by use of the Gauss-Jordan algorithm in subroutine form using an augmented matrix concept. Subroutines to perform matrix transposition, matrix multiplication and matrix times vector multiplication were also utilized.

Preliminary Preparations and Concepts

Digitizing of the control points in each map area utilizing the graphics workstation resulted in a file of coordinate data for each control point. One set of x,y coordinate data, expressed in State Plane Rectangular Coordinate values related to the digitization of apparent street intersections in the Parcel Centroid layer and the other represented the intersection node coordinates as provided in the GBF/DIME file for the relevent SMSA. Table 1 illustrates a short list of the structure of this file.

TABLE 1
CONTROL POINT COORDINATE FILE

Point	Census	Tract	Census	Base Map		GBF/DIME	
				Coordinates		Coordinates	
No.	Basic	Suffix	Node	X	Y	X	Y
1	5218	2	57	2155573.	163166.	2155596.	163088.
2	5218	2	48	2156517.	163041.	2156651.	162949.
3	5218	2	73	2156129.	162660.	2156099.	162508.
4	5218	2	81	2155836.	161957.	2155798.	161483.
5	5218	2	1103	2155289.	161537.	2155273.	161379.

The computer system developed to produce transformed composite maps was conceived to operate in an on-line, interactive mode and to permit as many iterations as desired based on selection of the number and distribution of the control points in the map area and on the choice of the transformation method. The production of parameters and consequent residuals on the control points is very fast on a Digital Equipment Corporation VAX 11/750.

Tests and Conclusions

As an illustration of the potential of utilizing global map transformation concepts, various tests were performed on a particular map area where the overlay of original GBF/DIME data on the base map was poor. Quick numerically oriented comparisons based on number of control points and transformation type produced results illustrated in Table 2. The results of this type of analysis (even before plotting) seem to indicate that a 2nd order polynomial would fit the particular map area best. A plot (Figure 2) of the area using the transformed GBF/DIME node data definitely shows great improvement over the original view.

Table 2 illustrates the tendency for the x,y residuals to increase as the number of control points chosen also increases. This is to be expected. In general, considering the randomness of the digitizing errors in the GBF/DIME system and also in the base system, it seems advisable to utilize a large number rather than a sparse number of control points. Taking out control points that have large residuals is also counterproductive as far as the final map results are concerned. The orthogonal similarity transformation produced the poorest solution as was expected considering the nature of the problem.

TABLE 2
CONTROL POINT RESIDUALS (M.S.E)
(in feet)

	Number of Control Points					
	8		10		20	
Transformation	Residual		Residual		Residual	
Type	X	Y	X	Y	X	Y
Linear Affine	24.5	18.0	25.8	20.1	31.6	26.9
Orthog. Similarity	27.4	19.6	26.6	20.7	36.1	36.5
Bilinear Polynom.	16.4	13.5	25.0	19.6	29.0	25.5
2nd Order Polynom.	15.7	12.4	23.9	14.4	24.3	23.6

One final point, the residuals must be made to vanish to avoid gaps or gores in the final product. The simple and effective solution utilized in this application was to force the GBF/DIME node at the control point to have the same x,y values as the digitized base point x,y values.

"RUBBERSHEETING" BASED ON TRIANGULATION

Development and Implementation of Delaunay Triangles

The preceeding approach of using global transformation parameters to fit one map into the framework of another (in this case GBF/DIME transformed to a Parcel Centroid base map) while yielding an improvement over the original untransformed composite, can still be improved by using a network of triangles connecting the control points in the map area. From the special analytic principles relating to the geometry of triangles, unique transformation parameters applicable to points only in each triangle, can be expected to result in better local correction of distortions instead of using a fixed set of transformation parameters considered applicable over the whole map area. The development of a computer solution for this methodology is more complex

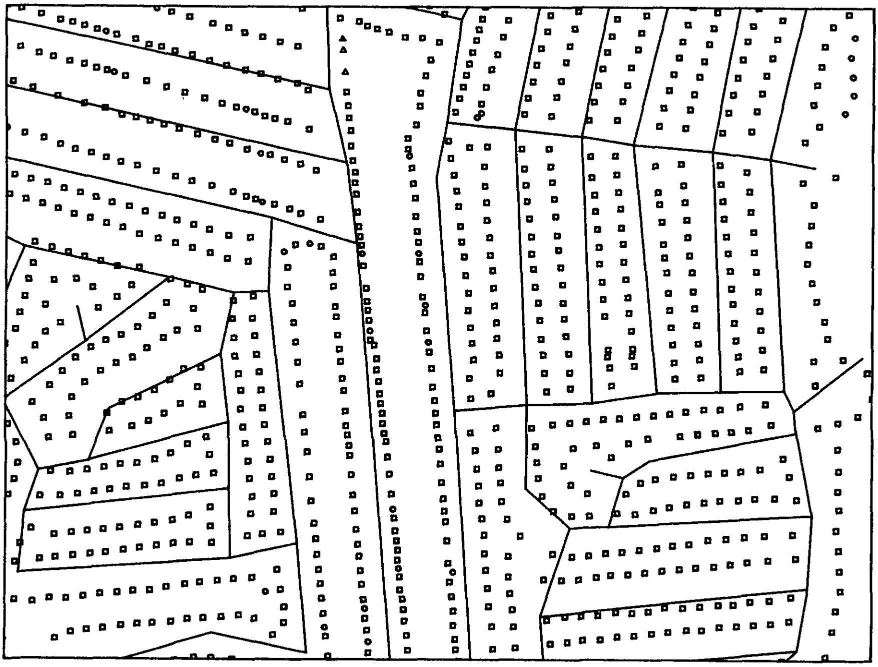


Figure 2
Illustration of Improvement by 2nd Order
Polynomial Transformation

than the previously discussed approach and will be described below.

Background
For some time now, Delaunay triangles (Delaunay 1934) and triangulation facets covering areas or surfaces have been found useful for contour mapping and in finite element analysis (Gold, Charters and Ramsden 1977; Elfick 1979; McCullagh and Ross 1980; Lewis and Robinson 1977; Brassel and Reif 1979). More recently the U.S. Bureau of the Census has focused attention on the advantages of using Delaunay triangulation to transform detail from one map into the domain of another map base (Gilman 1985, White and Griffen 1985; Saalfeld 1985).

Concepts and Algorithms Required
Many triangle arrangements are possible to connect a series of control points in a map area. Only a Delaunay triangulation is unique and independent of the starting point. Delaunay triangles are characterized by the fact that they contain no other control points. Thus in Figure 3, if points 1 and 2 determine a possible base of a triangle (point 2 is the closest neighbor to 1) then a new neighbor to the right of line 1-2 must be found such that the cir-

cumscribing circle passing through points 1 and 2 and the other new point contains no other data point to form a Delaunay triangle.

Obviously, the proper selection of the new point to form a triangle requires checking into possibly numerous candidates in the proximity of point 2. It is necessary to have an efficient proximity selection algorithm and many of these have been proposed in the literature (Friedman, Baskett and Shustek 1975). In the case of this program development, elements of the X and Y coordinates of the control points were encoded into a character string or integer word using the features of FORTRAN 77 available on the VAX. This scheme develops a one dimensional representation when sorted of the two dimensional map space of discrete points and emphasizes proximity relationships. Using this concept, relatively few points had to be tried to properly form a Delaunay triangle.

Besides the need for a proximity analysis, building a Delaunay triangle requires a point of view in terms of clockwise or counter-clockwise rotation about the focal point 1 in Figure 3. The direction concept chosen in this application was clockwise. Therefore, the new vertex point 3 must be to the right of the line 1-2 as well as a close neighbor to point 2.

The process used to check the direction of the selected point is based on the fact that the equation of a straight line in a plane

$$Ay + Bx + C = 0 \qquad (13)$$

divides the plane into two half planes such that points in one half plane satisfy

$$Ay + Bx + C > 0 \qquad (14)$$

and those in the other half plane satisfy

$$Ay + Bx + C < 0 \qquad (15)$$

For the clockwise point of view in Figure 3, a point will lie to the right of a line if the parameters are computed as:

$$A = (x_2 \; - x_1)/2(-1) \qquad (16)$$
$$B = (y_1 \; - y_2)/2(-1) \qquad (17)$$
$$C = (y_2x_1 \; - y_1x_2)/2(-1) \qquad (18)$$

and if Equation (14) is satisfied.

Substituting the x and y coordinates of the proposed point into the equation Ay + Bx + C, the point can be quickly rejected or accepted in terms of direction. The final review of the new vertex is done by a subroutine which incorporates a distance check with near neighbors.

When a Delaunay triangle is formed it must be entered into *a database structure* that allows easy retrieval of triangle elements and adjacency relationships. The simple system involved array storage of:

Triangle Number, 3 Vertex Point Numbers, Adjacent Triangles (up to 3), and 6 Triangle Transformation Parameters.

The method elected to move the GBF/DIME file triangle vertex coordinates into exact correspondance with the base map triangle coordinates (see Figure 4) involved the use of the linear affine equations discussed earlier in this paper:

$$X_1 = a_1x_1 + a_2y_1 + a_3 \tag{19}$$
$$X_2 = a_1x_2 + a_2y_2 + a_3 \tag{20}$$
$$X_3 = a_1x_3 + a_2y_3 + a_3 \tag{21}$$
$$Y_1 = b_1x_1 + b_2y_1 + b_3 \tag{22}$$
$$Y_2 = b_1x_2 + b_2y_2 + b_3 \tag{23}$$
$$Y_3 = b_1x_3 + b_2y_3 + b_3 \tag{24}$$

where:

X_1,X_2,X_3 are the final base map coordinates and
x_1,x_2,x_3 are the original GBF/DIME coordinates.

This set of six equations based on 3 points is just sufficient to derive the 6 transformation parameters for an individual triangle. The solution of the equations was done with the same Gauss-Jordan algorithm used in the earlier investigations. It is to be noted that this type of solution using the global coordinates contrasts with the use of "local" coordinate systems with their own concepts which has been favored by other investigators. It produces the same transformed results.

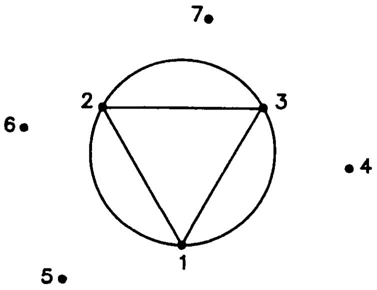


Figure 3. Creation of Delaunay Triangles

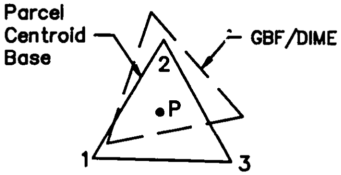


Figure 4. Generalization of Triangle Warpage

For the final transformation of the GBF/DIME nodes over the whole map area, each node point had to be identified as a point in a particular Delaunay triangle. To accomplish this the GBF/DIME nodes were placed in a file with an indexed structure which had the x,y coordinates encoded in the key similar to the earlier use in the point proximity algorithm described earlier.

By building an enclosing rectangle around each triangle and using the key structure to find a potential point, each point could be verified as being in a particular triangle by using the right of line testing method described previously. The triangle point transformations using the derived x,y transformation parameters was very quickly done. In this way each triangle, with its quota of points inside, was treated to provide the final total map area presentation in the base map system.

Summary and Conclusions

Figure 5 illustrates the type of solution achieved by using the triangulation approach. The scheme is particularly successful in handling layers of information including annotation in terms of preserving relationships for a pleasing composite map.

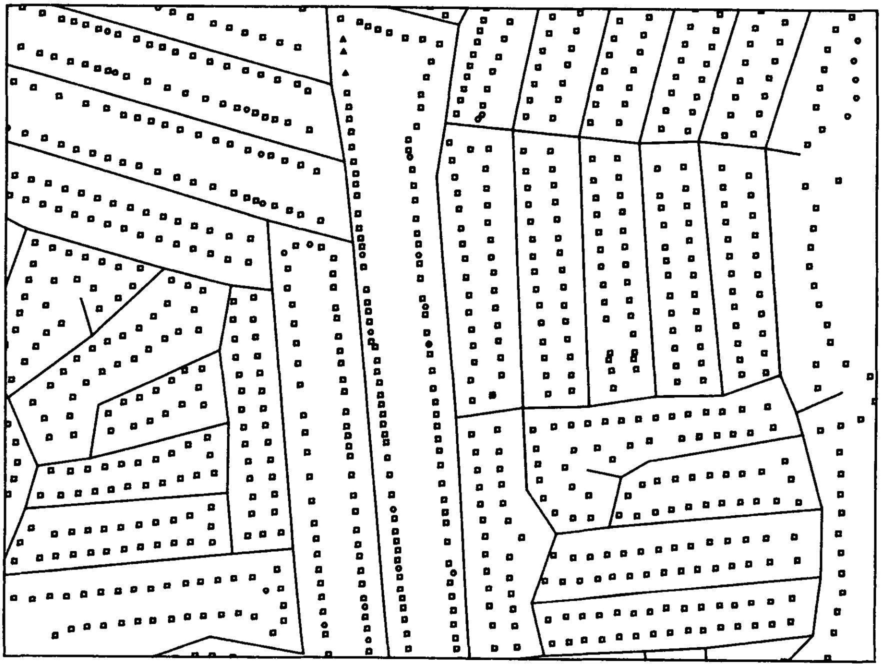


Figure 5.
Illustration of Improvement by Triangulation
Using Delaunay Triangles

REFERENCES

Bahr, H.P. 1976, Geometric Models for Satellite Scanner Imagery: ISP Commission III, Helsinki.

Brassel, K. and Reif, D. 1979, A Procedure to Generate Thiessen Polygons: Geographical Analysis, Vol. 11, No. 3, 289-303.

Delaunay, B. 1934, Sur la sphere vide: Bulletin of the Academy of Sciences of the USSR, Classe Sci. Mat. Nat., 793-800.

Elfick, M. H. 1979, Contouring by Use of a Triangular Mesh: Cartographic Journal, Vol. 16, June, 24-29.

Fagan, G. L. 1986, Routing of Municipal Service Vehicles, Town of Oyster Bay, Long Island, New York: Proceedings of the Annual Convention of the American Society of Photogrammetry and the American Congress on Surveying and Mapping, Volume 3, Washington, D.C.

Friedman, J., Baskett, F. and Shustek, L. 1975, An Algorithm for Finding Nearest Neighbors: IEEE Transactions on Computers, October, 1000-1006.

Gilman, D. 1985, Triangulations for Rubber Sheeting: Proceedings. AutoCarto 7, Digital Representations of Spatial Knowledge, American Society of Photogrammetry and the American Congress on Surveying and Mapping, Washington, D.C., 191-197.

Gold, C., Charles, T. and Ramsden, J. 1977, Automated Contour Mapping Using Triangular Element Data: Computer Graphics, 11, 170-175.

Kratky, V. 1972, Image Transformations: Photogrammetric Engineering, May, 463-471.

Lee, D. T. and Schachter, B. J. 1980, Two Algorithms for Constructing a Delaunay Triangulation: International Journal of Computer and Information Sciences, Vol. 9, No. 3, 219-242.

Lewis, B. A. and Robinson, J. S. 1977, Triangulation of Planar Regions with Applications: The Computer Journal, Vol. 21, No. 4, 324-332.

McCullagh, M. and Ross, C. G. 1980, Delaunay Triangulation of a Random Data Set for Isorithmic Mapping: The Cartographic Journal, Vol. 17, No. 2, 93-99.

Mikhail, E. M. 1976, Observations and Least Squares: Harper & Row, New York.

Saalfeld, A. 1985, A Fast Rubber-Sheeting Transformation Using Simplicial Coordinates: The American Cartographer, Vol. 12, No. 2, 169-173.

White, M. and Griffen, P. 1985, Piecewise Linear Rubber-Sheet Map Transformation: The American Cartographer, Vol. 12, No. 2, 123-131.

FIGHTING BUDWORM WITH A GIS

Glen Jordan
and
Leon Vietinghoff

Faculty of Forestry
University of New Brunswick
Fredericton, NB, CANADA E3B 6C2

ABSTRACT

This paper outlines the current state of GIS application in the ongoing battle against spruce budworm in the province of New Brunswick, Canada. In particular, the paper concentrates on describing a new methodology for applying GIS capabilities in the planning of aerial spray block layout. The paper concludes that the methodology has merit and is further evidence that current GIS technology does indeed offer the capability to move beyond simple retrieval applications into complex planning situations. However, slow response times due to large data volumes and current vector overlay approaches need to be addressed.

INTRODUCTION

Protection planning is an integral part of forest management. Forest management involves scheduling a host of activities associated with silviculture and harvest of the timber resource but, significantly, it also involves protecting the timber resource from insects, disease and fire. In the province of New Brunswick, Canada, the provincial Department of Natural Resources and Energy (DNRE) is responsible for forest protection on Crown (public) land and small freehold (private woodlots) land while forest companies pay for protection on their limits. The principal insect pest in the province, the spruce budworm (Choristoneura fumiferana (Clem.)), is the focus of their attention.

The spruce budworm is a prevalent and persistent insect pest over large regions of Canada and the United States. Budworm larvae annually defoliate spruce (Picea sp.) and fir (Abies sp.) trees over extensive areas in Central and Eastern Canada and, in the Northeastern States (Kettela, 1983). Spruce and fir are the mainstay of the pulp, paper and lumber industries in these regions. In New Brunswick these industries are the very backbone of the economy (Watson, 1983)! Repeated annual defoliation of spruce and fir trees by budworm leads to significant tree volume losses through slowed growth (Kleinschmidt, 1980) and, if unchecked, death in 3 to 5 years (MacLean, 1980.). New Brunswick has attempted to minimize such losses with annual aerial spray of pesticides. Conducted since 1952 (Irving and Webb, 1984), the spray programme has aimed to limit timber volume losses and maintain established levels of softwood harvests on Crown and small freehold land. It does not aim to eliminate the spruce budworm (Kettela, 1975).

Conducting an aerial spray campaign over an area the size of New Brunswick (approximately 6 million hectares of productive forest) is not a task that is accomplished without considerable planning effort. Actual spray operations are conducted by Forest Protection Ltd. (FPL), a non-profit company jointly owned by the provincial government and forest companies owning freehold land and holding Crown land licenses in the province. However, it is the Timber Management Branch of DNRE that actually plans the protection programme each year with input from those company sponsors having forest areas involved. Protection programme planning requires the collection and manipulation of large amounts of data, much of it map-based, in an effort to identify those forested areas needing protection and to configure these areas into operable spray blocks. The actual amount of forest targeted for aerial spraying each year varies with fluctuations in budworm populations. In 1986 approximately 500,000 ha were sprayed (Forest Protection Limited, 1986), though operations covering several million hectares have been common in the past 10 years (Irving and Webb, 1984). Until recently all data and map manipulation was carried out manually.

In 1982 DNRE purchased a GIS (ARC/INFO) with the intent of using it to store and handle forest and base map data for the entire province, and began the process of building the database (Erdle and Jordan, 1984). This massive task, almost complete, involves digitizing 2,000 forest cover-type maps (each covering approximately 4,000 ha at a scale of 1:12,500) and entering forest inventory and silviculture data for hundreds of thousands of forest stands. Once the database is complete, numerous applications of the GIS, particularly as a planning aid, will be possible. Already the GIS is being employed as an aid in protection programme planning and harvest scheduling (Erdle and Jordan, 1984). This paper will outline the current state of GIS application in the ongoing battle against the spruce budworm in New Brunswick. In particular, the paper will concentrate on describing current research aimed at developing a computer-based planning procedure for spray block layout.

THE PROTECTION PLANNING PROCESS

A number of map products are generated and used by DNRE in protection planning, including: (1) infestation forecast maps; (2) forest cover-type maps; (3) susceptibility maps; (4) setback maps; and (5) hazard maps. In the past, this information was generated each year and employed to target forest areas for aerial treatment in the form of spray block layout maps. A significantly different approach is now being attempted for the first time. The approach will arrange spray blocks solely on the basis of the distribution of susceptible forest and setback zones. Since the distribution of susceptible forest changes slowly over time, spray blocks, once arranged, will become "permanent" — subject only to minor adjustments from year to year. For actual treatment, spray blocks will be included or excluded, each year, on the basis of their coincidence with areas identified as being at risk due to past defoliation and predicted infestation. The paragraphs that follow outline the procedure in more detail.

The process of targeting specific forest areas for spraying begins with the identification of susceptible forest stands, i.e. those stands with characteristics that make them susceptible to budworm attack. Cover-type information, stored in DNRE's geo-referenced forest database, is used to produce 1:50,000 susceptibility maps. Each map, like the map illustrated in Figure 1, represents approximately 100,000 ha of ground area. The production of susceptibility maps is a relatively simple but time consuming undertaking involving the reclassification of basic stand information, such as species composition, into susceptible or non-susceptible categories on the basis of a specific set of rules.

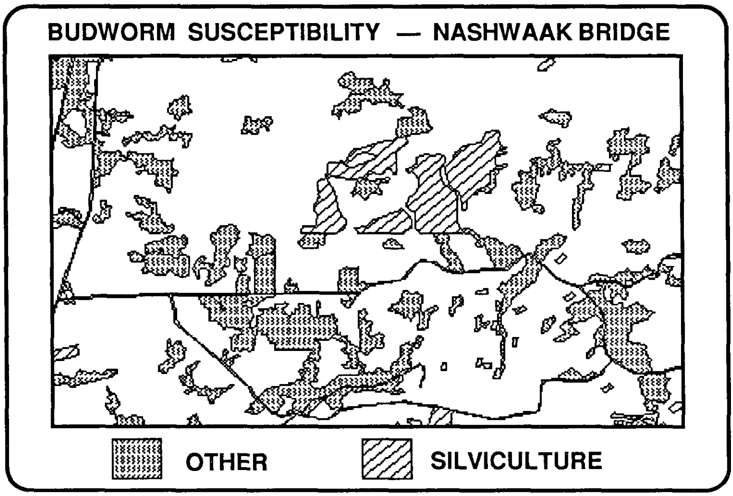


Figure 1. A 1:50,000 Map Showing Distribution of Budworm Susceptible Forest.

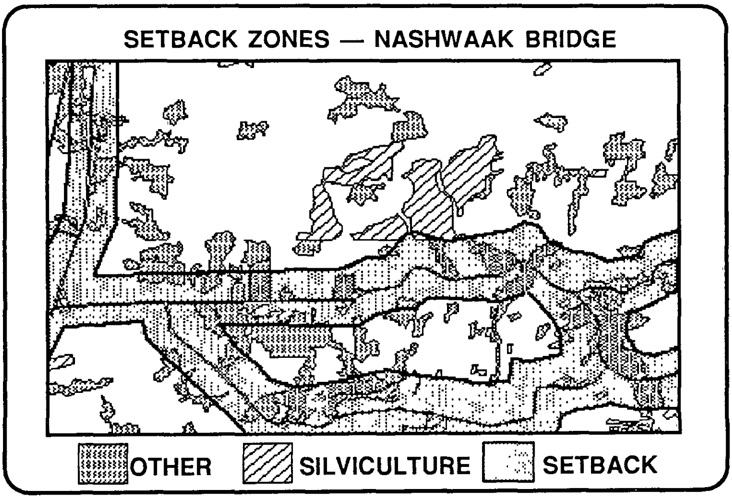


Figure 2. Budworm Susceptibility Map with Setback Zones Delineated.

The next step in the planning process involves identifying setback zones. Setback zones are buffers around habitation and ecologically sensitive areas which require special planning consideration in that they limit, or exclude altogether, aerial spray operations within their boundaries. Presently the delineation of setback zones is carried out entirely with manual mapping methods and results in a map typified by that shown in Figure 2. Although the GIS at DNRE has the ability to generate buffers around map features, the existing forest database does not contain the source information, for example the location of dwellings, fox farms or blueberry fields that would be necessary to generate setback zones. This situation is unlikely to change in the foreseeable future, since the cost to gather, digitally encode, maintain and process such information in a spatial database would be high and the payoff, versus current manual methods, apparently limited.

After setback zones have been outlined on susceptibility maps, the planning process enters one of its most difficult stages. This stage involves the layout of aerial spray blocks in such a way that areas needing protection are targeted in a configuration that will allow safe and cost-effective spraying. As might be guessed in looking at the map in Figure 2 this is not an easy undertaking, and in fact, it is not practical to target all parcels of susceptible forest. The process of laying out or modifying existing spray block configuration requires the integration of a host of considerations. For example, knowledge about aircraft types and associated load capacities, spray swath widths and relative operating costs, must be combined with visual impressions of spatial arrangements of parcels of susceptible forest and occurrence of setback zones and topographic features. The decision to locate and arrange a spray block in a certain way is subjective. The planner may readily see groupings of susceptible stands and configure a spray block accordingly; however, without a quantitative basis for evaluating a proposed spray block layout, the planner will always wonder whether a better configuration might exist. Figure 3 isolates a geographic area on a 1:50,000 susceptibility map and depicts two spray block proposals for the indicated analysis area. Each proposal appears, on the surface at least, to be as good as the other. Currently the quantitative basis for comparing alternatives does not exist. However, if a quantitative basis can be established, it would seem logical to use GIS technology as the basis of a planning tool for spray block layout, given the map-based nature of the problem and the existence of the geo-referenced forest database. The next section of this paper describes such an approach, currently being researched at the University of New Brunswick's (UNB) Faculty of Forestry.

The final step in targeting forest areas for spraying involves eliminating those spray blocks that are not coincident with forest areas at hazard. Hazard maps, presently produced manually by combining information on the location of forest areas subjected to repeated defoliation in recent years and predicted budworm population distribution for the coming year, are overlayed on spray block maps to isolate those spray blocks not containing areas at risk. (This last step would certainly be amenable to GIS application; however, this is beyond the scope of the project described in this paper.)

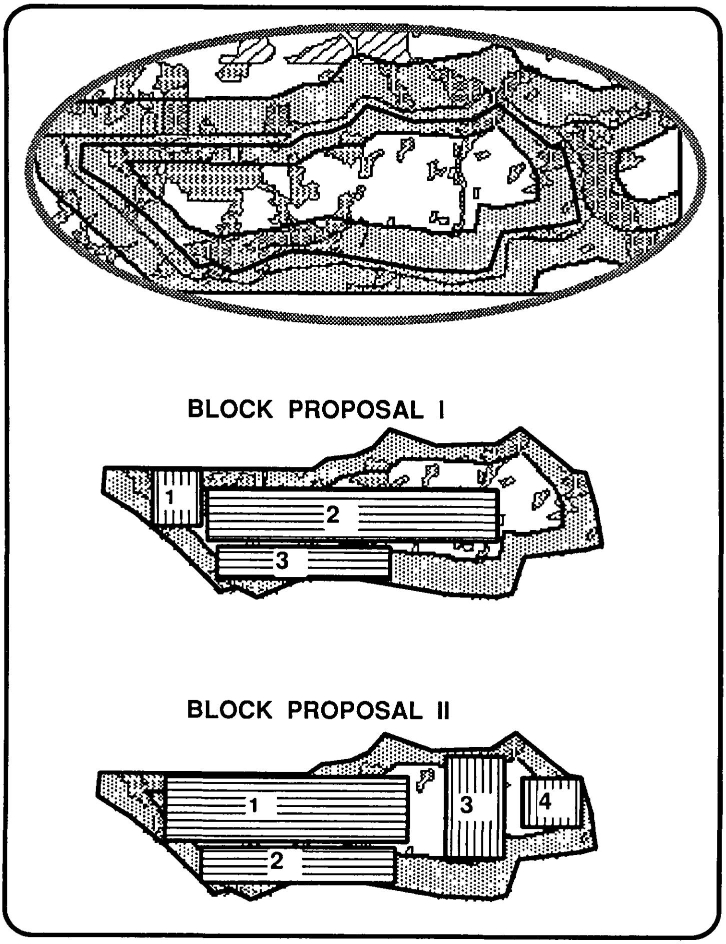


Figure 3. Ilustration of Alternative Spray Block Proposals for an Analysis Area.

A GIS-BASED APPROACH TO SPRAY BLOCK PLANNING

A prototype spray block layout procedure has been researched and programmed at UNB, with the support of both DNRE and FPL. The objective was to develop an iterative planning approach which would see the planner design alternative spray block configurations, communicate these to a computer for analysis, evaluate resulting quantitative feedback, and make further refinement to spray block layout as necessary. Thus the process was designed to be user-driven, taking advantage of the planner's ability to visually integrate spatial information from a map base, while using the computer for computation-intensive analyses.

The proposed spray block planning procedure described in the following paragraphs was developed and tested using the ODYSSEY GIS running under

TSO on UNB's IBM 3090 mainframe computer. A test database was constructed by extracting the forest cover-type data for 21 adjacent map sheets (approximately 90,000 ha) from DNRE's ARC/INFO-maintained forest database. This meant writing the data to magnetic tape in ASCII format on DNRE's PRIME 550 minicomputer and subsequently reformatting it on UNB's mainframe to create an ODYSSEY-compatible database. Applying actual DNRE rules for budworm susceptibility classification to maps in the test database permitted the creation of susceptibility maps, in digital form, for use in testing the proposed procedure.

The first step in the proposed spray block layout procedure is the definition of a "planning area" . Under present circumstances the planner works from a 1:50,000 map sheet base (91 maps in total) on which susceptible forest cover is highlighted over cultural as well as topographic features and elevation contours. Setback zones are also highlighted on these maps. Using the GIS-based planning procedure proposed, the planner will still work from the same 1:50,000 map base. Indeed, the planner will use this map base for all digitizing work as well. The planner must, however, identify those "digital" map sheets that define the planning area and then run a GIS procedure that digitally appends the identified maps to create one digital base map coverage of budworm susceptible forest.

The next step in the proposed planning procedure subdivides the planning area into component "analysis areas". These are non-overlapping areas which can be logically treated as separate entities for spray blocking purposes due to the presence of setback zones or vast areas of non-susceptible forest which form absolute or logical boundaries. Each analysis area is handled individually. The analysis area boundaries are digitized, and each analysis area in turn is extracted (via GIS overlay) from the underlying planning area. This allows quicker spray block proposal overlay in the next planning step.

The third step in the proposed procedure involves planning spray block layout, one analysis area at a time. Once the planner has designed a potential spray block configuration for an analysis area, it is communicated to the computer via digitizing of block corners. A digital overlay of these blocks (polygons) on the analysis area allows the generation of area statistics. For example, treatment costs can be calculated, based on spray aircraft type and area covered. Table 1 presents one possible cost summary for the two spray block proposals depicted in Figure 3. Note, that although average treatment cost per hectare of susceptible forest is slightly higher with proposal II, considerably more susceptible forest is targeted. In looking at Table 1, it is important to keep in mind that treatment costs using large aircraft (designated TBM) are considerably less, per hectare, than smaller agricultural-type aircraft; however, TBM aircraft are not allowed to spray chemical pesticides within the 1600 metre (1 mile) habitation setback zone, whereas small aircraft may operate within 300 metres of habitation. Other forest stand data, such as wood volume yields, could also be combined with costing data and incorporated in a tabular display. In any case, the statistics generated for one spray block configuration can be compared against alternative configurations or against a standard, indicating to the planner whether refinement of the configuration is desirable. The planner repeats the process with subsequent analysis areas until the current planning area is complete.

BLOCK PROPOSALS

PROPOSAL I	AIRCRAFT TYPE	TOTAL AREA (ha)	SUSCEPTIBLE AREA (ha)	COST
BLOCK1	AG	825	512	$ 9,751
BLOCK2	TBM	5,000	1,550	$ 36,150
BLOCK3	AG	1,800	864	$ 21,276
TOTAL		7,625	2,926	$ 67,177
	COST/HA SUSCEPTIBLE FOREST SPRAYED = $ 22.96			
PROPOSAL II				
BLOCK1	AG	5,100	2,550	$ 60,282
BLOCK2	AG	2,210	1,017	$ 26,122
BLOCK3	AG	1,860	744	$ 21,985
BLOCK4	AG	990	436	$ 11,702
TOTAL		10,160	4,747	$ 120,091
	COST/HA SUSCEPTIBLE FOREST SPRAYED = $ 25.30			

Table 1. Quantitative Comparison of Alternative Spray Block Proposals.

Other planning areas are identified and are processed as described until the entire province has been analyzed.

CONCLUSION

The spray block layout planning procedure outlined has been implemented as a prototype and tested using the ODYSSEY GIS at UNB. Work with the prototype indicates that GIS procedures can be usefully applied to the spray block layout problem. In particular, the prototype has shown that current GIS technology, programmed to capture and analyze spray block layout alternatives, has the potential to be a valuable tool in the complex process of protection planning. However, testing to date has raised concerns about implementation on a production basis. The concern stems from the fact that while both quick response and simplicity of use are deemed important, vector overlays are computation intensive and would certainly be slow to complete on DNRE's present computer system, assuming realistic data volumes. For example, using DNRE's PRIME 550 minicomputer (0.7 MIPS) the overlay of a spray block proposal involving a realistic analysis area of approximately 10,000 hectares (432 polygons, 773 arcs and 26,000 vertices), required in excess of 30 minutes of CPU time to complete. This response, although perhaps not surprising, is unacceptable. At the time of writing, means to improve response time, aside from hardware upgrading, were being researched.

REFERENCES

1. Erdle, T.A. and G.A. Jordan. 1984. Computer-based mapping in forestry: a view from New Brunswick. Canadian Forest Industries. 104: 38-46.

2. Forest Protection Limited. 1986. 1986 Program Report. Forest Protection Limited, P.O. Box 1030, Fredericton, N.B. E3B 5C3. 19 pp.

3. Kettela, E.G. 1975. Aerial spraying for protection of forests infested by spruce budworm. Forestry Chronicle. 51(4): 141-142.

4. Kettela, E.G. 1983. A cartographic history of spruce budworm defoliation from 1967 to 1981 in eastern North America. Env. Can., Can. For. Serv., M.F.R.C. Fredericton, N.B. Inf. Rept. DPC-X-14. 8 pp.

5. Kleinschmidt, S.M., G.L. Baskerville and D.S. Solomon. 1980. Reduction in volume increment in fir-spruce stands due to defoliation by spruce budworm. Faculty of Forestry, UNB, Fredericton, NB E3B 6C2. 37 pp.

6. Irving, H.J. and F.E. Webb. State of the art of "Forest Insect Control" in Canada as reflected by protection against the spruce budworm. Forest Protection Limited, P.O. Box 1030, Fredericton, N.B. E3B 5C3. 33 pp.

7. MacLean, D.A. 1980. Vulnerability of fir-spruce stands during uncontrolled spruce budworm outbreaks: an overview and discussion. Forestry Chronicle 56(5): 213-221.

8. Watson, R.S. 1983. New Brunswick forest industry and forest resources: an overview. New Brunswick Dept. Nat. Res., P.O. Box 6000, Fredericton, N.B. E3B 5H1. 30 pp.

AUTOMATION OF FLOOD HAZARD MAPPING BY THE FEDERAL EMERGENCY MANAGEMENT AGENCY

Daniel M. Cotter
Federal Emergency Management Agency
Federal Insurance Administration
Office of Risk Assessment
500 C Street, S.W., Washington, DC 20472

Daniel J. Lohmann
Greenhorne & O'Mara, Inc.
9001 Edmonston Road, Greenbelt, MD 20770

ABSTRACT

Flood hazard maps are currently produced by the Federal Emergency Management Agency (FEMA) with conventional cartographic methods. These maps, which depict areas that would be inundated by a flood having a one-percent probability of being equaled or exceeded in any given year (100-year flood), are produced to support the National Flood Insurance Program. FEMA is now evaluating techniques that can be used to automate the flood hazard mapping process, with the potential for developing an entirely computer-based system for the collection, analysis, and dissemination of flood hazard data. This paper presents the results of FEMA's initial efforts to automate topographic data collection and flood hazard map preparation.

INTRODUCTION

The Federal Emergency Management Agency (FEMA), an independent agency within the Executive Branch of the Federal Government, is tasked with administration of the National Flood Insurance Program (NFIP). Flood hazard mapping produced by FEMA provides the basis for NFIP community floodplain management, as well as flood insurance rate structuring. Since the inception of the NFIP in 1968, flood hazard areas have been mapped in 18,600 communities nationwide (Mrazik, 1986). Flood hazard maps depict areas that would be inundated by a flood having a one-percent probability of being equaled or exceeded in any given year (100-year floodplain), the 500-year floodplain, floodways, coastal high hazard areas, 100-year flood elevations, and insurance risk zones. Standard hydrologic, hydraulic, and modeling techniques are used to assess flood risks, and the resulting mapping is prepared through conventional methods. Paper map products are distributed by FEMA to NFIP communities, insurance agents, state agencies, and upon request, to any other interested party. During Fiscal Year 1986, the number of flood hazard map panels distributed by FEMA exceeded ten million.

The integration of developing technology in the fields of remote sensing and automated cartography into the NFIP can provide both economic and administrative benefits to FEMA and flood-prone communities. The collection of data required for risk assessment, particularly topographic data, is costly and time consuming. Paper maps, although functional, cannot provide the flexibility and analytical power of digital data incorporated into a Geographic Information System (GIS). Recognizing this, FEMA has developed a concept for automated flood study production (Mrazik, 1984). The purpose of this paper is to describe the Agency's initial progress toward fully automated flood hazard mapping (see Figure 1).

CONCEPT

Flood hazard studies are performed for FEMA by Study Contractors (SCs). The SCs may be private or public organizations with expertise in hydrologic and hydraulic analyses. The data required to support a flood hazard study include land cover, topography, hydrography, stream gage records, and cultural data. The results of SC analyses are displayed on maps at scales of 1:4,800 to 1:24,000. In a fully automated system, all data would be collected in digital format, integrated with existing digital base mapping, and analyzed through a GIS to produce flood hazard maps.

Initially, FEMA has identified two areas for experimentation as steps towards automating flood study production. These areas are the collection of topographic data (which represents, on the average, 35 percent of study production costs) and the feasibility of producing flood hazard maps with automated cartographic technology.

CONCEPTUAL DIAGRAM OF FULLY
AUTOMATED FLOOD INSURANCE STUDY PROGRAM

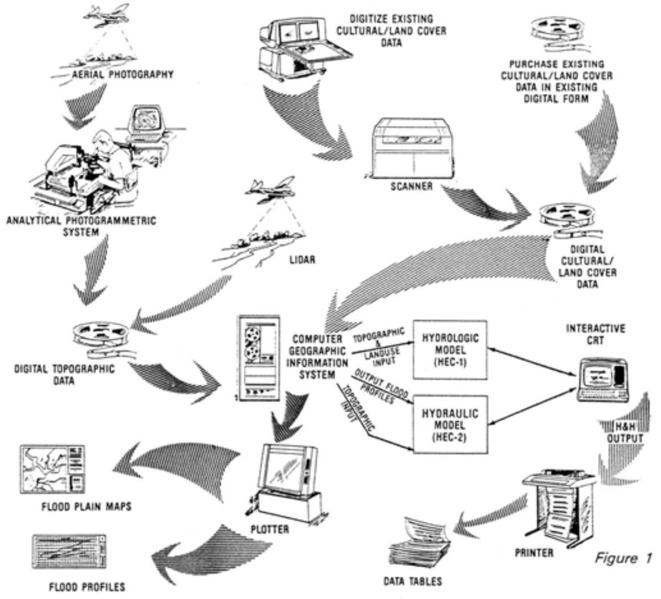

AERIAL PHOTOGRAPHY
DIGITIZE EXISTING CULTURAL/LAND COVER DATA
PURCHASE EXISTING CULTURAL/LAND COVER DATA IN EXISTING DIGITAL FORM
ANALYTICAL PHOTOGRAMMETRIC SYSTEM
SCANNER
LIDAR
DIGITAL CULTURAL/ LAND COVER DATA
DIGITAL TOPOGRAPHIC DATA
COMPUTER GEOGRAPHIC INFORMATION SYSTEM
TOPOGRAPHIC & LANDUSE INPUT
OUTPUT FLOOD PROFILES
TOPOGRAPHIC INPUT
HYDROLOGIC MODEL (HEC-1)
HYDRAULIC MODEL (HEC-2)
INTERACTIVE CRT
H&H OUTPUT
FLOOD PLAIN MAPS
PLOTTER
FLOOD PROFILES
DATA TABLES
PRINTER

Figure 1

AUTOMATED TOPOGRAPHIC DATA COLLECTION

Conventional photogrammetric methods are employed by FEMA in collecting much of the topographic data used for flood studies. In the past, photogrammetry has clearly been the most efficient means of acquiring these data. However, developing laser mapping holds promise for extremely rapid data collection regardless of leaf bloom, sun angle, or overcast conditions.

FEMA is currently participating with the Corps of Engineers in an operational field test of airborne LIDAR (Light Detection and Ranging Technology) (see Figure 2). The data collection portion of this test was conducted as part of FEMA's Hays County, Texas, flood hazard study. The test included comparison of LIDAR data with ground survey and photogrammetric data.

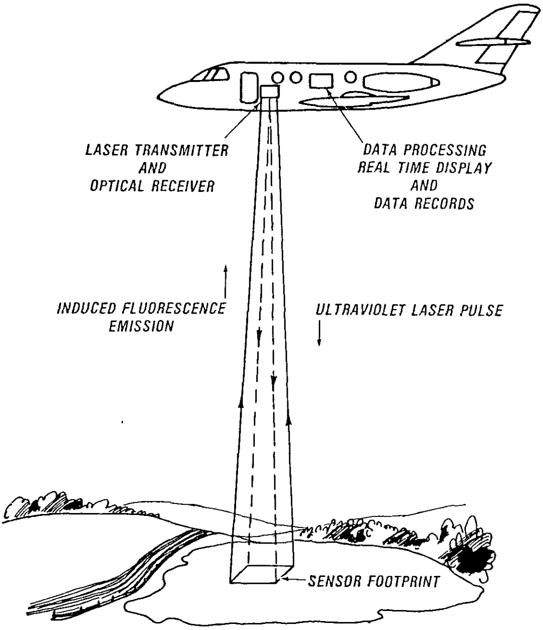

LASER TRANSMITTER AND OPTICAL RECEIVER
DATA PROCESSING REAL TIME DISPLAY AND DATA RECORDS
INDUCED FLUORESCENCE EMISSION
ULTRAVIOLET LASER PULSE
SENSOR FOOTPRINT

Figure 2

A full evaluation of the LIDAR system's accuracy, including the creation of a digital elevation model (DEM) has yet to be completed (Stole, 1986). Initial results indicated that a vertical accuracy of ±1.5 feet can be achieved with LIDAR. This accuracy is well within FEMA's requirement for four foot contour interval topographic mapping for flood hazard area identification.

AUTOMATION OF FLOOD HAZARD MAP PRODUCTION

At present, final flood hazard maps are prepared for publication by FEMA Technical Evaluation Contractors (TECs). These final products are based on work maps submitted by the SCs. After review of SC hydrologic and hydraulic analyses, the TEC transfers the flood hazard information to an appropriate base map. Conventional scribing, screening, masking, and photographic processes are used to prepare a final negative for printing.

The present format of flood hazard maps requires that two separate maps be prepared for studied areas. One map, the Flood Insurance Rate Map, or FIRM, portrays 100-year flood elevations, also referred to as Base Flood Elevations (BFEs), and insurance risk zones, while the second map, the Flood Boundary and Floodway Map, or FBFM, portrays the floodway (the portion of a floodplain set aside to convey the 100-year flood discharge without raising BFEs by more than 1.0 foot). As a result of a recent review (FEMA, 1985) of the FIRMs and FBFMs, FEMA has combined these two maps into a single map, which will display all flood hazard information in a simplified format. This review also called for the addition of horizontal control to flood hazard maps.

The addition of horizontal control is a significant step toward the creation of digital flood hazard mapping. Maps now published by FEMA lack horizontal control and are therefore difficult to digitize in correct spatial relation to the earth's surface. However, before flood hazard maps can be published with horizontal control, FEMA must establish a workable method to control these maps to a reasonable accuracy without a great cost to the mapping program.

During 1986, FEMA experimented with several methods of establishing horizontal control. The most effective method was found to be the transfer of horizontal control, in the form of geocoordinates, from U.S. Geological Survey (USGS) 7.5 minute topographic quadrangle maps to flood hazard map panels. An initial test program, using AUTOGIS*, showed that, in most cases, horizontal control can be transferred from the USGS quadrangle maps to flood hazard maps with an accuracy of about 0.1 arc second (FEMA, 1986).

Based on those results, FEMA is now engaged in a pilot project to determine whether this method of adding horizontal control to flood hazard maps is practical and cost effective for use in a production environment. Also of concern will be verification of the quality of the horizontal control. The pilot project will include independent checks of control points and internal map points to ensure that the maps are published with correct spatial relations relative to the earth's graticule.

Aside from determining a procedure for providing horizontal control, a digital data standard for flood hazard maps is required to facilitate automated mapping endeavors. A digital data standard for flood hazard maps must be acceptable to a wide variety of users, be well documented, and specify annotation codes as well as fonts. Further, the standard must be flexible so that unique flood hazard data can be incorporated with the attribute code specifications (see Figure 3).

* AUTOGIS is a flexible body of software packages, including AMS (Analytical Mapping System) and MOSS (Map Overlay Statistical System), for the analysis of spatial data developed by the Fish and Wildlife Service of the Department of Interior.

After reviewing existing standards, FEMA elected to adopt the USGS (DLG) format (USGS, 1985). The wide use of DLG and the design of USGS DLG feature codes to allow inclusion of non-standard map features, such as flood hazard data, were primary reasons for this selection. Figure 3 shows unique flood map symbols, and the USGS DLG codes for these symbols devised by FEMA. Other attributes, such as the base map transportation network and hydrography, are digitized and annotated according to appropriate USGS DLG standards.

The selection of USGS DLG, as amended to include the flood hazard map features shown in Figure 3, as a digital data standard is an important result of FEMA's initial efforts toward an automated mapping program. The Agency encourages users of FIRMs and FBFMs to apply these standards whenever flood hazard data are to be digitized for inclusion within a GIS.

TULSA, OKLAHOMA PILOT PROJECT

FEMA performed a pilot project during 1986 to test the process of digitizing flood maps, and to estimate costs associated with digital flood hazard map production, and to identify and resolve problems with the automation of flood hazard map production by contractors. For the pilot project, the existing flood hazard maps for Tulsa, Oklahoma, were selected (see Figure 4). These maps, originally published in 1971, presented a number of problems that would not normally be encountered in the digitizing process. Problems with these flood hazard maps included FIRMs and FBFMs at different scales, with different base map sources, and with different panelization schemes; lack of flood profiles or original flood insurance study text; poor readability of the FBFM (see Figure 4); and a lack of horizontal control.

SELECTED FEATURES, SYMBOLOGY, AND DLG CODING
UNIQUE TO DIGITAL FLOOD HAZARD MAPS

		DLG CODE	
FEATURE	SYMBOL	MAJOR CODE	MINOR CODE
Base Flood Elevations (with lines to be annotated with BFE)	∿∿∿∿∿∿∿∿∿∿	400	0001
Cross Section (Cross sections to be annotated by letter)	Ⓐ————————Ⓐ	400	0002
Floodway	—— —— —— ——	400	0003
Approximate A Zone (100-year flood) boundary	– – – – – – – – – –	400	0004
100-year Flood Boundaries	————————	400	0007
500-year Flood Boundaries	————————	400	0008
Zone D	——— ——— ———	400	0005
Gutters (zone boundary lines)	white lines ▨	400	0006

Figure 3

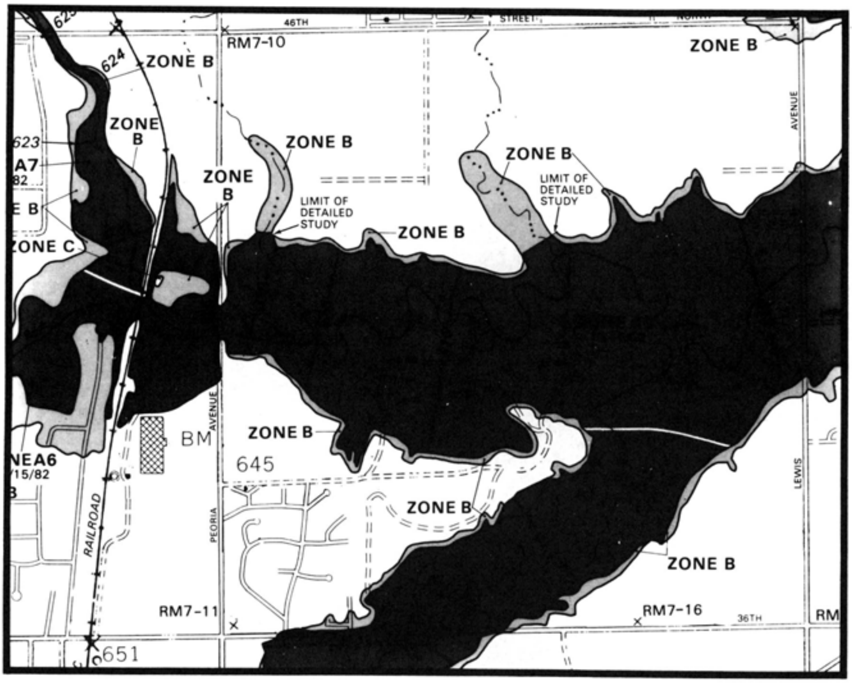


Figure 4

EXAMPLE OF PUBLISHED FIRM FOR THE CITY OF TULSA

In assessing the problems with the data set, several key tasks were identified for completion at the predigital stage. The primary problem, requiring considerable manual effort, was preparation of mylar overlays of data contained on the FBFM. These mylar overlays were keyed to the FIRM panels for digitizing. This step, which would not normally be necessary in digitizing flood hazard maps, was required because of the poor readability of the FBFM (see Figure 5).

Digitizing was performed using a manual system (Intergraph). Data were initially recorded as a continuous string, with data being captured by thematic topics. Quality control of the digitized data was achieved by comparison of an intermediate map output, or "check plot" of digitized data with the original map. Correction averaged 10 to 30 percent of the entire digital data set per map. The error level varied depending on the operators' experience and the complexity of the original geometry digitized. Quality control and edit were found to require between two and four times the amount of time required to simply digitize data.

The digitized data were converted to USGS-Digital Line Graphics Level 3 (DLG-3) through the creation of nodes and attribute coding of the digital data set. Software available on Intergraph allowed much, but not all, of this task to be automated. Intergraph DLGIN/DLGOUT software was used to create and convert files from Intergraph format to DLG and vice versa. Some difficulties were encountered in this procedure resulting from limits formerly inherent in Intergraph software relating to the number of points, nodes, and areas that can be identified in a given data set, and the expansion of the DLG-3 codes to include flood data.

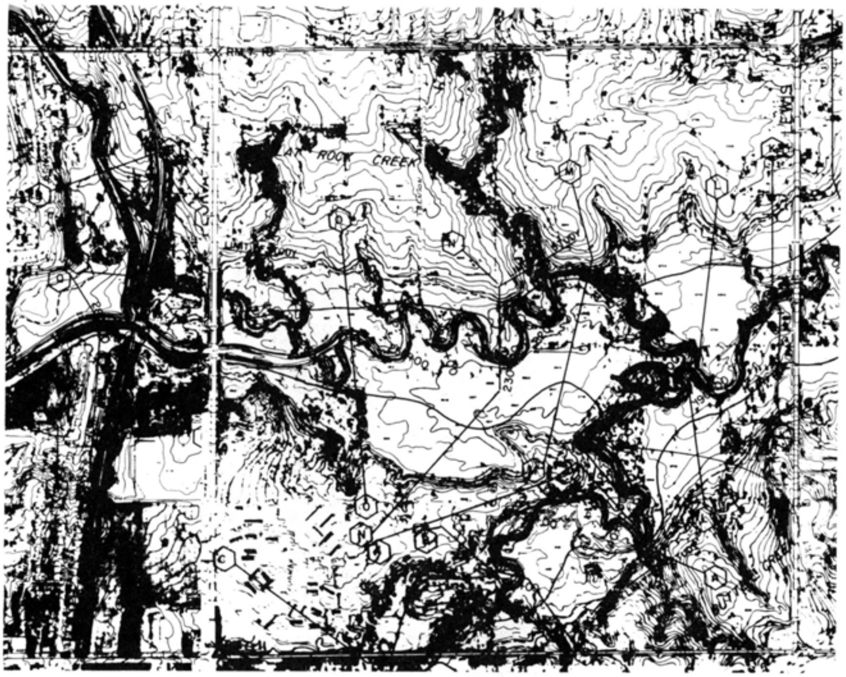


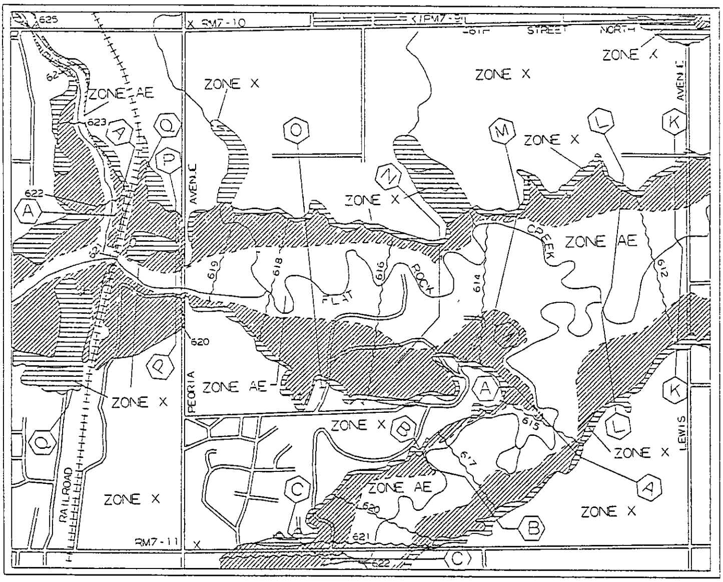
Figure 5

EXAMPLE OF PUBLISHED FBFM FOR THE CITY OF TULSA

The resulting digital data set for Tulsa is a data file in a topological format, designed to be integrated with GISs. The data were captured in relation to geocoordinates. The DLG-3 output tape is in USGS Standard Distribution Format, a direct charter representation (ASCII) of the binary file translated from the Intergraph system.

Graphics output using the digital data set and various plotting devices was generally found to be comparable with conventional flood hazard map graphics (see Figure 6). However, three areas require further research:

1) text placement;
2) removal of road casings at intersections ("cleanout"); and
3) duplication of the screen used for the 100- and 500-year floodplains.



EXAMPLE OF COMPUTER-GENERATED
FIRM FOR THE CITY OF TULSA

Figure 6

TIME AND COST ANALYSIS

In evaluating the time required to apply automated cartographic techniques to produce flood maps, compared with that to produce flood maps through conventional procedures, it was estimated that the time required to produce flood hazard maps in an automated mode could increase production time requirements by a factor of four; and the cost to produce flood hazard maps would increase by a factor of two if automated cartographic technologies were applied.

Some of this unfavorable comparison of computer vs. conventional mapping is the result of selecting a worst-case map set for the pilot project. Considerable pre-digital manual effort was expended that would not normally be required. However, even given that this resulted in a high time-and-cost figure for automated map production, it is not likely that a simple case would provide a much more favorable comparison. In part, this is because the Agency relies heavily on USGS 7.5 minute quadrangle map separates for base map material. Manual flood hazard map production requires only that these existing base map data be photographically modified to the correct scale and the flood hazard data be overlain. The automated process requires the time consuming and costly creation of a digital data set containing the base map information.

A more viable option would be that only the data related to flood hazards be digitized by FEMA in the USGS DLG format, and that no base map information be recorded. The cost of pursuing this option is an increase in FEMA flood hazard map production costs of about 40 percent above those currently experienced. This relates to a cost of about 800 dollars to produce digital flood data for a single FEMA flood hazard map panel. Acceptance and use of such a data set would be dependent, to a large extent, upon the existence of digital base map data and the willingness of users to create their own digital base map information.

CONCLUSIONS

As a result of this project, two conclusions are clear:

1) It is technically possible to produce high quality flood hazard maps from digital data; and
2) The increase in flood hazard map production costs that would result from a conversion to automated flood hazard map production is unacceptable to FEMA unless benefits to the tax payer can be identified outside map production that will justify the increased cost.

A third conclusion must also be drawn from this project: Given the versatility of GIS technology, FEMA should perform a benefit analysis to determine if sufficient value can be assigned to the use of digital flood hazard data by other Federal agencies, as well as state, local, and private organizations to justify the creation of a digital flood hazard data base by the Agency.

FUTURE DIRECTION

In the future, FEMA expects to continue the assessment of LIDAR technology, and to incorporate this technology into some flood insurance study data collection efforts. Developing technology in the field of remote sensing, particularly as it applies to topographic data collection, will be monitored by FEMA for its applicability to NFIP requirements.

FEMA has also developed a GIS that can provide on-line capabilities for the analysis of spatial data. This system, the Integrated Emergency Management Information System (IEMIS), is intended to provide low cost access to digital data, planning models, information management, and networking capabilities. IEMIS data are structured in a digital line graph format, and the system has the ability to read in data structured according to USGS DLG specifications. Further information on IEMIS can be obtained by writing to: Federal Emergency Management Agency, State and Local Programs Directorate, Technological Hazards Division, Washington, DC 20472.

IEMIS will be a cornerstone of the assessment of benefits that would result from the creation of a digital flood hazard data base. The Agency will base the benefit analysis on the following assumptions:

1) Only flood hazard data produced by FEMA will be digitized by the Agency;
2) USGS 1:100,000 scale maps, now being digitized by the Bureau of Census, can serve as adequate base maps for digital flood hazard data;
3) Both the digital flood hazard data and the digital USGS 1:100,000 scale data can be made available to users through FEMA's IEMIS, which will also provide on-line GIS capabilities to users; and
4) Digital flood data will only be generated for communities with significant populations and properties at risk from flood hazards and a significant demand for data in digital form.

BIBLIOGRAPHY

Federal Emergency Management Agency, Methodology for Adding Horizontal Control to Flood Maps, Washington, DC, September 30, 1986 (unpublished agency document).

Federal Emergency Management Agency, Map Initiatives Project, Final Report, FEMA-FIA-ORA, Washington, DC, January 1985.

Mrazik, B. R., Applications of Mapping Technology to Flood Insurance Studies, Association of State Floodplain Managers, Proceedings, Portland, Maine, June 1984.

Mrazik, B. R., Status of Floodplain Hazard Evaluation Under the National Flood Insurance Program, American Institute of Hydrology, Washington, DC, September 1986.

Stoll, J. K., Status Update of Airborne Laser Topographic Survey Demonstration, Transportation Research Board, Committee A 2A01, Workshop Report, Santa Fe, New Mexico, July 1986.

U.S. Geological Survey, National Mapping Program Technical Instructions, Standards for Digital Line Graphs, Part 3, Attribute Coding, Reston, Virginia, July 1985.

EXPERT SYSTEMS APPLIED TO PROBLEMS IN GEOGRAPHIC INFORMATION SYSTEMS: INTRODUCTION, REVIEW AND PROSPECTS

Vincent B. Robinson
(Goss.Ensuadmin@UNCA-MULTICS.MAILNET)
Department of Surveying Engineering
The University of Calgary
2500 University Drive, NW
Calgary, Alberta   T2N 1N4
CANADA

Andrew U. Frank
(Frank@Mecan1.BITNET)
Department of Civil Engineering
University of Maine at Orono
Orono, ME 04469
USA

ABSTRACT

This paper discusses the nature of expert systems with special attention on construction of expert systems. We identify four major problem domains of geographic information systems in which expert system technology has been applied - map design, terrain/feature extraction, geographic database management, and geographic decision support systems. Efforts in each problem domain are critically reviewed. Considering the accomplishments and shortcomings of efforts to date, we suggest areas for future research. Two areas in particular need of further consideration are methods of knowledge acquisition, and formalization of both knowledge and uncertainty.

INTRODUCTION

In previous papers we introduced expert systems for land information systems (LIS) (Robinson et al 1986b), critically surveyed efforts related to expert systems for geographic information systems (GIS) and identified several research themes for developing expert system technology for GIS (Robinson et al, 1986c,d). In this paper we direct more attention to expert system construction. This topic was not presented in detail in previous papers and is typically ignored by those developing expert systems for applications in LIS, GIS and automated cartography. We then proceed to provide a critical update to our previous surveys (Robinson et al, 1986c,d). Considering trends in the field and the evolution of our thinking, we elaborate on various aspects of expert system research and development of particular importance to GIS.

One may think of expert systems as computer systems that advise on or help solve real-world problems requiring an expert's interpretation. They solve real-world problems using a model of expert human reasoning, reaching the same conclusions that the human expert would reach if faced with a comparable problem (Weiss and Kulikowski, 1984). For a more detailed introduction to expert systems readers are referred to Robinson et al (1986b).

CONSTRUCTING EXPERT SYSTEMS

Generally speaking, expert systems go through a number of stages that closely resemble classical systems analysis - identification, conceptualization, prototyping, creating user interfaces, testing and redefinition, and knowledgebase maintenance. Also, it has been observed that once the thrill of a prototype system and a fancy interface wears off, many projects come to an abrupt end as the expense of developing them further and maintaining them is assessed (Bobrow et al, 1986).

Identification

To identify problems amenable to solution through expert system technology, a critical mass might be one or two knowledge engineers and a group of experts. Five to ten test cases should be collected for later use. With distributed knowledge, the interview process should expose specializations and the degree of consensus in solution methods among the group of experts.

Conceptualization

Once the domain has been identified the next step is conceptualization and formalization of knowledge. Initial knowledge acquisition sessions should start with a single expert who can demonstrate by working through several examples. Having developed some sense of the problem the knowledge engineer can then begin to articulate in a semiformal language what is believed to be going on in the problemsolving sessions.

A useful next step is simulating the process of solving of one or more test cases. After several rounds of simulation by knowledge engineers critiquing by single expert, it is often useful to bring in other experts to help identify idiosyncracies and determine the multiplicity of problem-solving styles. In the Pride project (Mittal et al, 1985) knowledge acquisition sessions led to creation of a "design knowledge document." It outlined different stages of design, dependencies between stages, and provided a detailed rendering of various pieces of knowledge (rules, procedures, constraints, etc). Before the first line of code had been written the document had evolved to 20+ closely typed pages with 100+ pages of appendices. It reportedly played a crucial role in defining and verifying knowledge eventually incorporated into the Pride system. It was circulated among experts for comment, correction, and identification of omissions. Thus, it helped make explicit some of the knowledge that had been implicitly applied by experts.

User Interfaces

One of most important and time consuming stages in developing expert systems is creation of suitable user interface. Particularly one that matches what users of the noncomputer system have been accustomed to. Goal browsers are an artifact of the user interface unique to expert systems. These goal browsers can be used to lay out the expert system design process as a network of different goals and displays goal status during the construction stage. They also sometimes allow the user to edit, undo, advise and reexecute goals.

Testing and Redefinition

Once a prototype has reached the stage where it is possible to go through the initial test problems from beginning to end it becomes important to start testing the system with friendly users. This usually reveals new problems. Thus, it is common for a second or even a third version of a prototype may be developed. Feedback from solving real problems often forces reimplementation - a cycle characteristic of knowledge programming.

Knowledgebase Maintenance

After friendly users have tried the system a plan must be developed for a large software development project. The plan must provide for testing, development, transfer, and maintenance of the knowledgebase. A process must be put in place at user locations to help tune the user interface, and extend the knowledgebase as new problems are found and easier ways to interact with the system are suggested. When the plan is complete one can more easily evaluate the cost of resources required versus the value of solving problem.

SOME EFFORTS IN EXPERT SYSTEMS AND GIS

There have been a number of expert system efforts reported that are relevant to GIS problems. Table 1 illustrates the relationship between problem domains of geographic information systems and activities particularly applicable to expert system development. The problem domains are : (1) automated map design and generalization, (2) terrain/feature extraction, (3) geographic database management and (4) geographic decision support.

We note a number of reported efforts in Table 1 that we do not discuss here. Some have been discussed in our previous papers, such as MAP-AID (Robinson and Jackson, 1985), ACES (Pfefferkorn et al, 1985), and ACRONYM (Brooks, 1983), while others are not reported in sufficient detail or have been abandoned recently, such as CES (Muller et al, 1986). Here we limit our discussion to a select group of efforts relevant to the exploitation of expert system technology to improve state-of-the-art in GIS and LIS.

Map Design

MAPEX is a rule-based system for automatic generalization of cartographic products (Nickerson and Freeman, 1986). This system was designed to work with USGS 1:24,000 DLG data being generalized to 1:250,000. Like other efforts in this field, there as no effort to extract expertise from human experts in map generalization. However, a significant contribution of this effort has been the formalization of the problem of generalization within a rule-based framework and the identification of existing rules and generation of rules-of-thumb. It is worthy of note that MAPEX was developed at the same institution that developed AUTONAP.

Table 1. Some Expert System Efforts Relevant to GIS Problem Domains.

Problem Domain	Expert System Effort
Map Design	
General	MAP-AID, MAPEX, CES
Name Placement	AUTONAP, ACES
Terrain/Feature Extraction	Palmer, ACRONYM, FES, CERBERUS, MAPS, SPAM
Geographic Database Management	LOBSTER, SRAS, KBGIS-I, KBGIS-II, ORBI, Wu
Geographic Decision Support	TS-Prolog, URBYS, DeMers GEODEX

AUTONAP (Ahn, 1984; Freeman and Ahn, 1984) is perhaps the most successful name placement expert system developed to date. This system emulates an expert cartographer in the task of placing feature names on a geographic map. However, like MAPEX there was no reported effort in extracting knowledge from an expert in name placement.

Terrain/Feature Extraction

Palmer (1984) showed how logic programming can be used as the basis of an expert system for analysis of terrain features. Using a triangular tesselation he represented nodes with their elevation, segments and triangles as first-order predicates. Then using Prolog to conduct symbolic analyses he demonstrated how valleys, streams, and ridges could be detected using the procedural knowledge encoded in a knowledge base and using Prolog control mechanisms. This work was subsequently extented by Frank et al (1986) to illustrate how physical geographic definitions might be formalized using logic programming methods.

FES is a Forestry Expert System (Goldberg et al, 1984) used expressly to analyze multi-temporal Landsat data for classification of landcover and landcover change of interest to foresters. Using a multi-temporal Landsat image database, production rules are applied in two phases. First production rules are used that involve change detection inference coupled with a reliability measure. The second phase generates decision rules regarding the current state of the image. The control structure of FES has been described as a "feedforward" system without backtracking.

CERBERUS was developed initially at NASA for the purpose of performing unsupervised classification of Landsat mulitspectral data (Engle 1985). It is data-driven rather than goal-driven. This FORTRAN-based system is currently being sold for $ 1750 through Cosmic as a knowledgebased system for experimenting with expert systems (Digital Review, 1986: 188).

Geographic Database Management

ORBI is an example of an expert system implemented in Prolog. It was developed to keep track of environmental resources for the country of Portugal. There are aspects of a classification system for environmental data and a decision-support system for resource planning. ORBI provides (1) a natural language parser for Portuguese that supports pronouns, ellipses, and other transformations, (2) menu handler for fixed-format input, (3) an explanation facility that keeps track of the steps in a deduction and shows them on request, and (4) help facilities that explain what is in the database, the kinds of deductions that are possible, and the kinds of vocabulary and syntax that may be used (Pereira et al, 1982). It remains one of the most impressive accomplishments todate.

LOBSTER (Frank, 1984), like ORBI, is based on the logic programming paradigm. It is a new implementation of a task previously solved using a traditional programming approach, namely a query language for a geographic database (Frank ,1982). It serves as an intelligent user interface to a spatial database managment system using the network data model rather than the relational model. It is felt that the flexibility in building the interface using a Prolog-like language was significant.

Smith and Pazner (1984) reported a prototype KBGIS that makes extensive use of several vintage methods drawn from the field of artificial intelligence. The objective of this system appears to have been to illustrate the use of techniques of artificial intelligence for search and simple learning on a spatial database. However, like so many other efforts, the last significant publication on this KBGIS reports it is under complete revision (Smith and Pazner, 1984).

Glick et al (1985) provide a more comprehensive design for a KBGIS using what they call hybrid knowledge representation. In contrast to Smith and Pazner (1984), who chose data structures that fit easily into the scheme of discrimination nets, Glick et al (1985) chose to use a variety of representation methods. Also reported is the use of a frame-based semantic net to represent the "meaning" of geographical objects and their interrelationships provides the capability to incorporate new entities, attributes, and relationships into the KBGIS.

Wu and Franklin (1987) describe an algorithm for polygon overlay that is implemented in Prolog. We include this work because of the importance of the polygon overlay problem to geographic database management and their use of Prolog to formalize the process of polygon overlay. This work in consistent with our suggestion that increased formalization of geographic knowledge be pursued (Robinson et al, 1986 a,b,c).

SRAS (Robinson et al, 1986d) is a spatial relations acquisition station. It is concerned with acquiring representations of natural language concepts to be used in subsequent queries of a geographic database. This is an mixed-initiative, question-and-answer system that chooses questions based on anticipated user response and its effect on the representation of the NL concept. It is one of the very few efforts in acquiring representations from 'experts' rather than developing rule-bases. Another unique feature of this effort is its recent concern with the composition of multiperson concepts for subsequent use in expert systems (Robinson and Wong, 1987).

Geographic Decision Support

Barath and Futo (1984) describe a system for comparing requirements of economic sectors and social factors. This goal-directed system is based on TS-Prolog. TS-Prolog is Prolog extended to allow for parallel processes and system simulation. Even though a simple example is presented, it is not clear whether the system is capable of using existing databases. This system also remains at the level of experimental applications, primarily funded by the Ministry of Industry of Hungary. Finally, like most of the above systems, the user interface has been given scant attention.

URBYS (Tani, 1986) is an expert system to aid in territorial planning and analysis of urban areas. Although there is recognition of the need for formalizing planning knowledge, it is nunclear whether the rigors of expert system construction will be followed in the elaboration of URBYS. Its organization is characteristic of the hybrid systems. Rather significantly, there is no formal provision for knowledge acquisition. It is left to the "expert" to change the rules and/or facts.

GEODEX (Chandra and Goran, 1986) was built to assist planners in evaluating site suitability for landuse activities. Its rules are drawn informally from a landuse planner. Using rules in its knowledgebase, GEODEX operates in a forward-chaining fashion applying site constraints drawn from the knowledgebase. There is mention of a capability of backtracking should the constraints prove so restrictive over the geographic database that no sites satisfy the constraints. As with most other systems, GEODEX is still under development.

DeMers (1986) reports an effort to develop a strategy for acquiring knowledge from landuse experts for use in the Land Evaluation and Site Assessment system. DeMers (1986) did not link knowledge acquisition to a formal method of knowledge representation, therefore little distinguishes this study from a multitude of other planning studies.

FUTURE PROSPECTS AND RESEARCH NEEDS

Robinson et al (1986a,c) suggest that many of the areas of past efforts will continue to be areas of research. For example, the map design problem will continue to be a focus of expert system development activity. However, it will increasingly make use of spatial databases. Given the research priorities of major funding agencies in the United States, terrain/feature extraction will continue as a very active area of expert system development. Geographic database management is quite a broad field and has implications for all the other research fields. Use of logic programming appears to be one of the more predominant trends which suggests that deductive geographic databases may become available soon. Much future work in spatial data error analysis, data capture and storage, and data transfer will be conducted within the context of this research theme. Development of spatially distributed databases containing data from a wide-variety of sources will encourage development of expert systems that navigate through a distributed system, combine contents of different databases, determine reliability of information, and maintain semantic variations.

Formalizing Knowledge

What is most notable about the efforts currently underway or proposed is that there is little concern for the process of knowledge acquisition and representation. Future developments of demand formalization of knowledge domains previously left partially formalized. These domains include cartographic design, terrain analysis, geomorphological feature extraction, extraction of natural and man-made features. This includes formalizing the process of knowledge acquisition and representation, something lacking in almost all of recent systems.

Formalizing Uncertainty

Much recent research in the field of artificial intelligence and expert systems concerns one of the byproducts of knowledge formalization - the formalization of uncertainty (Lesmo et al, 1985). It is clear that as progress is made in expert system development the importance of managing uncertainty will increase. For example, FES (Goldberg et al, 1984) included a reliability measure, Shine (1985) has reviewed the utility of Bayesian, Fuzzy, and Belief logics in feature extraction systems, and Robinson (1986) has reviewed the implications of fuzzy logic for geographic databases.

CONCLUDING COMMENTS

Given current computing infrastructure, expert systems are likely, over the near-term, to remain largely research/experimental systems. Developments are most likely to follow those already emerging. Prototypes will play an extremely important role in the future of this field. Prototypes based on a formal language of artificial intelligence will not only bring practical results but, more importantly, formally explore some of geography's less formalized areas. However, we feel that little will be contributed unless these prototypes are based on a rigorous approach to knowledge acquisition and representation. Thus, we feel that attempting to build a prototype expert system could be justified just on the amount of insight gained in the process of building it.

Recently Bobrow et al (1986) suggested that problems known to require a predominance of commonsense knowledge, english-language understanding, complicated geometric/spatial models, complex causal/temporal relations, or the understanding human intentions are not good candidates for current state-of-the-art expert systems. Most if not all of these problems are central to development of practical geographic expert systems. Thus, we suggest that there is much basic research to be done before practical geographic expert systems become a reality.

ACKNOWLEDGEMENTS

We would like to acknowledge partial support of a research grant from PSC-CUNY and National Science Foundation Grants IST-8412406 and IST-8609123.

REFERENCES

Ahn, J.K. 1984, Automatic Map Name Placement System, IPL-TR-063, Image Processing Laboratory, Rensselaer Polytechnic Institute, Troy, NY.

Barath, E. and I. Futo. 1984, "A Regional Planning System Based on Artificial Intelligence Concepts," Papers of the Regional Science Association, 55 : 135-154.

Bobrow, D.G., S. Mittal and M.J. Stefik. 1986, "Expert Systems: Perils and Promise," Comm ACM, 29(9), 880-894.

Brooks, R.A. 1983, "Model-based Three-dimensional Interpretations of Two-dimensional Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5, 140-150.

Chandra, N. and Goran, W. 1986, "Steps Toward a Knowledge-based Geographical Data Analysis System," in B. Opitz (ed.) Geographic Information Systems in Government, A. Deepak: Hampton, VA.

DeMers, M.N. 1986, "A Knowledge Base Acquisition Strategy for Expert Geographic System Development," in B. Opitz (ed.) Geographic Information Systems in Government, A. Deepak: Hampton, VA.

Digital Review. 1986, New Software product report on CERBERUS, p. 188.

Engle, S.W. 1985, CERBERUS Release Notes, Version 1.0, NASA Ames Research Center, Moffett Field, CA.

Frank, A.U., B. Palmer, and V.B. Robinson. 1986, "Formal Methods for Accurate Definitions of Some Fundamental Terms in Physical Geography," Proceedings 2nd Intern'l Symp on Spatial Data Handling, Seattle, WA.

Frank, A.U. 1984, "Extending a Network Database with Prolog," First International Workshop on Expert Database Systems, October, Kiawah Island, SC.

Frank, A.U. 1982, "MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation", Computer Graphics, 16, 199-207.

Freeman, H. and Ahn J. 1984, "AUTONAP - An Expert System for Automatic Map Name Placement," Proceedings 1st Intern'l Symp. on Spatial Data Handling, 544-571.

Glick, B., S.A. Hirsch, and N.A. Mandico. 1985, "Hybrid Knowledge Representation for a Geographic Information System," Paper presented at AutoCarto-7.

Goldberg, M., M. Alvo, and G. Karam. 1984, "The Analysis of Landsat Imagery Using an Expert System: Forestry Applications," Proceedings AutoCarto-6.

Lesmo, L., L. Saitta, and P. Torasso. 1985, "Evidence Combination in Expert Systems," Int. Jrnl. Man-Machine Studies, 22, 307-326.

Muller, J.-C., R.D. Johnson, and L.R. Vanzella. 1986, "A Knowledge-Based Approach for Developing Cartographic Expertise," Proceedings 2nd Intern'l Symp on Spatial Data Handling, Seattle, WA.

Nickerson, B.G. and H. Freeman. 1986, "Development of a Rule-Based System for Automatic Map Generalization," Proceedings 2nd Intern Symp on Spatial Data Handling, Seattle, WA.

Palmer, B. 1984, "Symbolic Feature Analysis and Expert Systems," Proceedings Intern'l Symp on Spatial Data Handling, 465-478.

Pereira, L.M., P. Sabatier, and E. de Oliveira. 1982, *ORBI - An Expert System for Environmental Resource Evaluation through Natural Language*, Report FCT/DI-3/82, Departmento de Informatica, Universidade Nova de Lisboa.

Pfefferkorn, C., D. Burr, D. Harrison, B. Heckman, C. Oresky, and J. Rothermel. 1985, "ACES: A Cartographic Expert System," *Proceedings* AutoCarto-7.

Robinson, V.B. and Frank, A.U. 1985, "About Different Kinds of Uncertainty in Geographic Information Systems," *Proceedings* AutoCarto-7.

Robinson, V.B., A.U. Frank, and M.A. Blaze. 1986a, "Expert Systems Applied to Problems in Geographic Information Systems: Introduction, Review, and Prospects," *Computers, Environment, and Urban Systems*, (in press)

Robinson, V.B., A.U. Frank, and M.A. Blaze. 1986b, "Introduction to Expert Systems for Land Information Systems," *Jrnl Surveying Engineering*, 112(2): 109-118.

Robinson, V.B., A.U. Frank, and M.A. Blaze. 1986c, "Expert Systems and Geographic Information Systems: Review and Prospects," *Jrnl of Surveying Engineering*, 112(2): 119-130.

Robinson, V.B., M.A. Blaze, and D. Thongs. 1986d, "Representation and Acquisition of a Natural Language Relation for Spatial Information Retrieval," *Proceedings* 2nd Intern'l Symp on Spatial Data Handling, Seattle, WA.

Robinson, V.B. and R.N. Wong. 1987, "Acquiring Approximate Representations of Some Spatial Relations," *Proceedings* AutoCarto-8, in press.

Robinson, G. and Jackson, M. 1985, "Expert Systems in Map Design," *Proceedings* of AUTOCARTO-7, 430-439.

Shine, J.A. 1985, "Bayesian, Fuzzy, Belief: Which Logic Works Best ?," *Proceedings* ASP, 676-679.

Smith, T.R. and M. Pazner. 1984, "Knowledge-Based Control of Search and Learning in a Large-Scale GIS," *Proceedings* Int. Symp. on Spatial Data Handling, 2, 498-519.

Tanic, E. 1986, "Urban Planning and Artificial Intelligence: The URBYS System," *Computers, Environment, and Urban Systems*, 10(3/4), 135-146.

Weiss, S.M. and Kulikowski, C.A. 1984, *A Practical Guide to Designing Expert Systems*, Rowman and Allenheld, Totawa, NJ.

Wu, P.Y.F. and W.R. Franklin. 1987, "Polygon Overlay in Prolog," *Proceedings* AutoCarto-8, in press.

THE EXPERT GEOGRAPHIC KNOWLEDGE SYSTEM
APPLYING LOGICAL RULES TO GEOGRAPHICAL INFORMATION

Kenneth D. Gardels
Landscape Architecture Computer Applications Laboratory
University of California
Berkeley, California 94720

ABSTRACT

The Expert Geographic Knowledge System (EGKS) represents the merger of techniques of computerized expert systems with those of geo-processing and database management. It involves the application of heuristic rules developed by experts in land management and related disciplines to the data within a geographic information system. EGKS construction must conform to rigorous design criteria to ensure that the system is capable of addressing the variations in the planning domain, that expert knowledge is accurately codified into rules reflecting its complexity and uncertainty, and that textual and graphic information is meaningfully communicated to the user.

INTRODUCTION

Environmental planning comprises the detailed information, rigorous analysis procedures, creative design and synthesis capability, and communication facilities necessary to understand and manage the relationship between humans and their environment. Planning practice relies on the application of expertise by specialists and generalists to the environmental decision-making process. The application of expertise to geographic information is the underlying concept of the Expert Geographic Knowledge System (EGKS).

The EGKS, using a computer and accessing large geographic and textual data bases, works in much the same way as the human expert. It applies logical rules stored in a knowledge base to the data in one or more geographic information system databases and to a large textual data base (called a domain cyclopeadia) in order to provide textual answers to specific geographic inquiries.

THE EGKS ARCHITECTURE

The goal of the expert geographic knowledge system is to provide expertise to an environmental specialist, a planner, or other decision-maker on the subject of a site, event, or topic specific inquiry. The overall architecture reflects the relationship between typical planning issues and the codification of expert knowledge in a computerized form. A planning problem emphatically does not involve the derivation of a specific statement from a limited set of facts. The converse is true: the planner must synthesize a wide realm of knowledge, and isolate that which is necessary to make the decision at hand. Thus the expert computer system must selectively utilize all the information available to it to deduce the information the decision-maker needs. It must be flexible enough to address a wide range of demands and meaningfully communicate its conclusions.

The architectural model of the EGKS is as follows: The user provides the system the basic parameters of the inquiry such as the location of a proposed development, the circumstances of an environmental event, or concerns about a resource issue. With this data, the expert system performs a complete investigation of relevant automated geographic information. Based on what it finds, the expert system applies rules established by human experts to make conclusions about the inquiry - the nature of the problem and realistic solutions to it. The system then assembles textual information relevant to the inquiry and produces documents detailing its findings. Finally, the planner or decision-maker can interact with the system for more knowledge or explanation.

DESIGN CRITERIA FOR AN EGKS

The construction of an expert geographic knowledge system requires a systematic approach to design. Design criteria must be rigorously applied and rationales for each standard must be explicitly defined for each component of the EGKS.

> The inference engine must be able to address the class of problems represented by environmental planning while retaining a high degree of domain independence for specific applications.
>
> The knowledge base must adequately reflect the complexity of planning and specialist expertise while remaining internally consistent and logical.
>
> Conclusions and explanations derived by the knowledge base rules must be supportable by general domain knowledge contained in an environmental cyclopeadia.
>
> The link to the GIS and other databases must retrieve the exact information necessary to address an inquiry.
>
> The output must be in formats useful to specialists, planners, and decision-makers and sufficiently flexible to accommodate varying needs.
>
> The user environment must be comfortable and encourage productivity while providing adequate power and capability to service advanced expertise requirements.
>
> The system must be amenable to updating and expansion in an open-ended, incremental fashion as new knowledge critical to landscape analysis becomes available.

Standards for ensuring that these basic requirements are met are necessary for any system design, whether domain independent or application specific.

The EGKS Inference Engine

The inference engine is the body of software code that translates the rule statements into actual actions and produces conclusions or other results. It uses the computational ability provided by list processing software to apply high-level rules to variable data.

Domain Independence. The key criterion for the inference engine is that it function independently of the facts of the domain(Hayes-Roth, Waterman, and Lenat, 1983) and remain adaptable to divergent sets of domain facts. For the EGKS, a workable inference engine must consider the *types* of relationships between correlated environmental data and the analytical and heuristic approaches used to study them, without being tied to specific relationships that exist only in one application or one location. Domain independence ensures technical compatibility between multiple systems, in terms of languages, coding, and so forth. Database adaptability ensures that systems can be implemented for new expertise need situations without rewriting the entire package. For a proposed application, these two factors mean that a system that meaningfully addresses local environmental issues can be developed fairly quickly.

Logical independence of the domain also implies a logic path that is complete and is not biased toward specific classes of solution. This may be described as *due process* reasoning, since it is based on both advocacy and skepticism. It has the distinct advantage in a planning domain of not being adversely affected by the absence of data that could be critical to a problem (Hewitt, 1985).

Antecedent-Driven Reasoning. Most expert systems are rule-based production systems, meaning that pattern matching, scheduling of rule-firing, and other logical operations are under the explicit control of an executive program. Within this framework, systems may be either forward or backward chaining, or antecedent-driven or consequent-driven, respectively (Infotech, 1981). If rules are defined as having two parts, an antecedent (the "if" part) and a consequent (the "then" part), the distinction between forward and backward chaining becomes a matter of whether any identified true antecedent produces its associated consequent(s) or any desired consequent is evaluated by testing its antecedent(s).

Most of the expert functions of the EGKS require a forward-chaining inference algorithm, so that a wide range of possible scenarios can be explored starting from the basic data. Thus the system is free to draw any reasonable conclusion from the data rather than seek out a particular conclusion (or diagnosis). This deductive process corresponds to the way in which geographic data, both manual and automated, are typically used, ie., data driven rather than goal driven. As intermediate hypotheses are derived, additional conditions may need to be established to verify a conclusion. At this point, the system may initiate backward-chaining logic in an attempt to determine if the relevant conditions are supported by the data.

Blackboard Hypothesis Interaction. A type of forward-chaining system that provides some of the desired data and goal driven functionality is the blackboard model, in which intermediate hypotheses are "posted" for examination by other rules. The blackboard-type procedure is closely analogous to traditional planning, where answers develop slowly and iteratively. The blackboard always represents partial analysis in which islands of truth begin to appear until its contents are complete and resolved (Waterman, 1981). By posting results to the blackboard complex interactions of basic environmental phenomena can be tracked and then reevaluated in terms of new knowledge and data.

One body of rules may be considered specialists and form hypotheses from basic data for posting on the blackboard - for example, the existence of a sensitive waterfowl habitat. Specialist rules are selected ("fired") based on activation rules that specify which rules, based on their content and their certainty, can best address the current best hypothesis on the blackboard - for example, to confirm or deny the existence of a wetland-associated soil type. Finally, strategy rules choose the activators that correspond to the class of knowledge needed to answer an inquiry - for example, to determine if a wetland/habitat area is of relevance to the study issue.

The blackboard can also point to slates containing relevant data extracted from the GIS or other databases, cyclopeadic information about the utilization of particular rules, and user-volunteered information about a site or event. As new information becomes available, or as new parameters are introduced, the blackboard will dynamically adjust to the new environment, and the program will select and fire new rules, reject old hypotheses and propose new ones, and backtrack and eliminate incorrect lines of reasoning.

Explanation. Most expert system packages being used today provide explanation to the user in the form of rule restatements. While explanation is important to users of any type of expert system, including the EGKS, the expert geographic knowledge system architecture expands the explanation capability significantly via the domain cyclopeadia. Material expertly extracted from the cyclopeadia explains the significance of each conclusion, not just the logical path used to reach it. Moreover, obtaining cyclopeadic materials based on content analysis of the text isolates the cyclopeadia from the rule-making and reasoning process. This allows the cyclopeadia to be revised, incrementally expanded, and updated without any effect on the inference engine or the knowledge base rules.

The EGKS Knowledge Base

The performance of an expert system is most closely related to the content of the knowledge base. Shridharan notes that the key to a thorough knowledge base and an expert level of performance is "formalizing, structuring and making explicit the private domain of a group of experts" (Infotech, 1981).

Knowledge Engineering. The acquisition and codification of expertise is the function of knowledge engineering. It involves identifying both the organization of the domain and the strategies of domain problem solving. The expert geographic knowledge engineer understands the nature of the environmental planning domain and the capabilities and constraints of the EGKS architecture. He or she is thus able to translate the heuristics of expert analysis and decision-making into the set of rules comprising a planning knowledge base. The knowledge engineer also defines the logical reasoning framework for the utilization of the rules - priority ordering, finding and hypothesis and conclusion building, etc. The duality of content and structure is especially important to environmental planning, since professional expertise consists largely of reasoned explanation and description rather than logical conclusion.

The knowledge engineer constructs a working EGKS by mapping the formalized knowledge into the representational framework provided by the blackboard/antecedent-based EGKS engine. For example, a general rule such as "Steep slopes with clayey soils tend be be unstable when wet" may become:

If slope | 12%,
and if soiltype = fine clay,
then potential slope instability is high.

The use of a high level rule-writing language allows rules to be stated in a restricted natural language format, and thus rules can be back-checked against the original experts' heuristic problem solving methods. Since codifying of reasoning used in environmental analysis involves restating ad hoc general rules of thumb into much more precise language, these restatements should be carefully reviewed with domain experts to ensure their applicability and internal consistency. The knowledge engineer and domain expert together postulate rule credibility indicators to indicate the relative reliability of each rule.

EGKS Rule Content. Although rules are by definition high-level expressions of domain expertise, each rule in a domain knowledge base should be relatively primitive. Attempting to express too much in a rule reduces the overall certainty of that rule, and as individual rules approach the universality of large-scale models they lose the incremental, hypothesis-building advantages of the expert system. For example, attempting to add to the previously cited slope stability rule considerations of nearness to upslope water sources, drainage nets, or precipitation rates would reduce the overall credibility of the original if-then statement. Limiting the scope of rules does not mean that individual rules cannot have multiple antecedents or consequents or numerical expressions of validity. However, each rule in the knowledge base must be simple enough that a single concept is represented - one that can be empirically confirmed or denied by research and field investigation.

Rules should also address important user questions and information needs. Extraneous or marginal rules reduce the surety of fundamental domain rules because, in the forward-chaining model, each activated rule that proves true from the database is combined with other rules proved true from other data to form intermediate and high-level hypotheses.

All cited rules become part of the final explanation or conclusion, and thus multiple rules incorporate multiple database elements as well. Therefore, the final conclusion reflects reductions in reliability as a function of map overlay, as well as the lowest common denominator of each individual rule's reliability.

Finally, rules should adequately represent both explicit "textbook" planning knowledge and the intuitive, experiential knowledge implicit in expert analysis. It is the latter that produces environmentally sound advice in a complex problem-solving or decision-making context. The knowledge engineer will probably have the greatest difficulty in building an EGKS knowledge base at the level of substantiating these heuristic rules. The point to be made is that they are the rules used in traditional approaches to planning, and their insertion into an automated system does not in itself lessen or enhance their status. However, leaving them out of an EGKS knowledge base severely restricts the degree to which the system can emulate a human expert and provide meaningful information.

Logical Reasoning Paths. Rules must express not only facts of the environmental planning domain, but must also direct the interaction of other rules. Just as the planning process involves relative weighing of multiple pieces of information and assigning of priorities, the expert geographic knowledge system architecture requires rules that direct the logical flow of knowledge from database to user. Although rule consequents point directly to other rule antecedents, a strategy must be imposed to ensure that the data based process is directed toward a class of explanation or conclusion. As described above in relation to the blackboard, this prioritizing is handled via strategy rules that reduce the solution space. Strategy rules point out additional data sources or initiate a query to the user when more information is needed to formulate or confirm a hypothesis.

Basically, a rule is a statement in the form "if *a*, then *b*." *a* may be a primitive comprising geographic data extracted from a GIS map, or it may be in effect the *b* of another rule. Both the *a* and *b* of a rule may have several components as well. Thus the rule-implementing process quickly becomes a network of conditions contemporaneously interacting on the blackboard. At any one time, a set of basic data, findings about those data, and conclusions in the form of hypotheses and explanations may all be represented. The inference engine in combination with high-level strategy rules is responsible for reducing the blackboard knowledge to a series of expert statements about the environment that resolve the original problem and address the user's inquiry.

Ultimately, the reasoning, blackboard updating, antecedent checking, and consequent firing process must reach an end. That is, the system must reach a stable state in which its expertise has produced conclusions about the environment that remain unchanged unless new data are provided. At this point, the expert system can communicate the statements as findings or explanations of the environment relative to the original inquiry.

The Domain Cyclopeadia

The EGKS domain cyclopeadia is the repository of non-rule knowledge in the specific environmental planning application. It provides the user with an organized summary of knowledge related to the findings of the rule evaluation process. Cyclopeadic materials that explain findings and elaborate on their decision-making significance may be expertly retrieved by the EGKS based on descriptors of material content. By being thus indirectly keyed to hypotheses and findings, the system can also extract knowledge explaining the means and results of its advanced deep reasoning processes. Based on application needs, the cyclopeadia may contain a variety of information types and be organized according to a range of structures.

The content of the domain cyclopeadia in the EGKS architecture comprises digests of knowledge relative to specific planning problems. The material included in the cyclopeadia serves various explanation and description functions. It can justify individual rules by describing the environmental relationships between an antecedent and consequent; for example, the relationship between mapped soil type or slope and the potential for slope instability. It can describe strategies and the hierarchies of rule applications, including procedures followed by environmental scientists in determining relevant resource parameters; for example, based on a finding of soil instability, what confirming indicators, such as vegetation or geology, are used. Thus it can amplify the situations defined by hypotheses and conclusions far beyond that expressed by a rule-series consequent. Most importantly, knowledge represented in the cyclopeadia can incorporate recommendations based on laws, statutes, or other land regulatory jurisdiction, not just on the EGKS reasoning process. For example, if the knowledge base rules determine that an environmental hazard exists at a site and that development should minimize adverse effects through use of setback zones and engineering restrictions, the cyclopeadia can explicitly define what those zones or restrictions should be and substantiate them in terms of precedence or jurisdictional authority.

A useful structure for representing knowledge within the cyclopeadia is to define each bit of organized knowledge as a <u>kernel</u> comprising text and/or graphics, each identified by keywords. The process of retrieving cyclopeadic knowledge is a function of identifying the kernels describing a rule or conclusions. Knowledge sources are mapped to keywords, and kernels satisfying the requirement are extracted and ordered. Keyword-based retrieval is in itself an expert process, since each rule, finding, description, and so forth must have some means of identifying additional cyclopeadic information relevant to itself. This function translates into a series of expert rules of the form, "if conclusion <u>a</u>, then find knowledge about subject <u>b</u>". Of course, rules may have complex interactions that build and reject hypotheses concerning what knowledge is relevant, just as the basic environmental rules do.

The Geographic Information System Link

The knowledge-based rule and cyclopeadia architecture rests fundamentally on basic geographic information contained within an automated database. Therefore the design criteria for the GIS data, their format, and their accessibility by query is critical to the operation of the expert geographic knowledge system.

Standards for data accuracy and currency are taken as *a priori* requirements for a GIS. Beyond these standards, though, are more exacting specifications to ensure that data obtainable from the GIS are capable of supporting expert analysis represented in the knowledge base. To be truly useful, GIS data must correspond to the accuracy and scale of user inquiries, and the data classification to the environmental information that is key to analysis and planning. Thus every feature on the automated map is useful, needed delineations are all present, and unnecessary data are minimized.

The format of the GIS data is also important to system performance, though, like the data, it may be out of the hands of the system designer or knowledge engineer. Because of the high-level rule structure of the EGKS, it is preferable that the GIS be accessible via fairly high-level calls at an operating system or query language level. GIS query should comprise a functional description of the mapped data (eg., natural vegetation and elevation province) rather than a structural description of the GIS organization (eg., columns 5-7 and 12). Otherwise, database information must be built into the EGKS and any GIS update, change, or expansion requires a major effort. A relational data model (Blumberg, 1975) and a modular software toolbox of functional data manipulation capabilities is critical to efficient data retrieval (Dangermond, 1983).

EGKS Output

The expert geographic knowledge system is designed to provide documented expertise to the environmental planner (or specialist or other decision-maker) about a site proposal, an environmental event, or other land-related issue. Expertise may be disseminated in a user-specified document containing the findings reached by the system, substantiated by textual and graphic material from the cyclopeadia. Optionally, maps of selected geographic features may be incorporated into the document as well. System expertise may be obtained interactively: the user directs queries to the system following the completion of an expert review of a proposal, event, etc. The system uses its logical record and information available on the blackboard to explain its reasoning, extract additional information, or even modify its conclusions based on new data.

The EGKS User Agent

The expert geographic knowledge system is designed to interact with planning professionals, not computer operators or programmers. By eliminating the data manager link for routine automated geographic queries, users are given much freer access to their data. Such direct contact requires that the system be easy to operate, recover gracefully from errors, and provide significant amounts of assistance to the novice user. Computer systems developed in the last few years, especially in the personal computer realm, have introduced the concept of a user agent that stands between the user and the actual operating system or program. An expert user agent not only makes an expert system easier to use, it understands the types of queries being made of it and can thus interpret application needs more accurately.

The EGKS user agent is the mechanism for translating initial user inquiries into the expert procedures used to reach conclusions and the means by which the system conveys those responses back to the user. The most complex part of the user agent is concerned with the interactive inquiry review - requesting explanation, eliciting information, or resetting parameters. For explanation requests, the agent displays the rules producing a specific finding. For information requests, the user chooses from a list of available topics to obtain more information from the database or cyclopeadia; for example, soil types in an area and their suitability for construction. For the parameter resetting - or "what-if" - function, the user agent conveys the causal relationships between parameters and findings by displaying the current parameter set as defined on the blackboard. A change in any parameter (or addition of a new one) results in a new reasoning cycle and the development of new conclusions. The new findings may be displayed on the screen and a map updated, using shading or color, to show new or changed areas of concern and to facilitate straightforward comparisons of different scenarios.

Continuing Knowledge Acquisition. The process of knowledge acquisition does not end after the initial knowledge engineering phase. On-going interaction with experts and incremental addition to the knowledge base and cyclopeadia are assumed. Moreover the system expertise can be directed toward acquiring new knowledge on its own. Interactive expertise transfer programs guide the expert in explicating and formalizing his or her knowledge.

Continued knowledge acquisition programs are particularly important to the EGKS because of the size and open-endedness of the environmental planning domain. The EGKS must consistently track its own decisions and determine where its reasoning is inconsistent with decisions made by human experts. Where this occurs repeatedly, it must identify the invalid assumptions or missing rule logic and postulate a remedy. The goal is a systematic, incremental implementation of EGKS technology and knowledge that accurately assesses the complex realm of environmental planning.

CONCLUSION

Expert systems represent the evolution of computer-aided decision-making from sequential number crunching to advanced reasoning, and as such represent the cutting edge of computer applications to real-world problems. The expert geographic knowledge system applies expertise-based reasoning capabilities to environmental planning problems - land use and management, environmental protection and monitoring, resource utilization and assessment - to identify impacts of and alternatives to development, to describe environmental effects of various activities, and to explain complicated resource issues.

The sophistication of the EGKS concept requires an equally sophisticated inference engine as well as data, information, and knowledge bases that are complete, accurate, current, and consistent. Where this is done, planning and decision-making in complex natural and institutional environments can benefit enormously.

REFERENCES

Blumberg, M. H. 1975, A Generalized Geographic Information System for Managing a Non-Redundant Demographic and Resource Data Base: Computers, Local Government and Productivity, Volume I, O. M. Anochie, ed., p. 181-193, URISA, Chicago. (Papers from the Thirteenth Annual Conference of the Urban and Regional Information Systems Association, August 24-28, 1975, Seattle.) IBM, Federal Systems Division.

Dangermond, J. 1983, A Classification of Software Components Commonly Used in Geographic Information Systems: Proceedings, United States/Australia Workshop on Design and Implementation of Computer-Based Geographic Information Systems, D. Peuquet and J. OCallaghan, eds., p. 70-91, IGU Commission on Geographical Data Sensing and Processing, Amherst (NY). Environmental Systems Research Institute.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. 1983, Building Expert Systems, Addison-Wesley Publishing Company, Inc., Reading (MA). (Teknowledge Series in Knowledge Engineering.) Teknowledge, Inc.; The Rand Corporation; Stanford University.

Hewitt, C. April 1985, The Challenge of Open Systems: Byte, 10(4): 223-244. MIT Artificial Intelligence Laboratory.

Infotech 1981, Machine Intelligence (collected papers), Pergamon Infotech Limited, Maidenhead, Berkshire, England. 407 pages.

Waterman, D. A. 1981, Rule-Based Expert Systems: Machine Intelligence, Infotech, ed., p. 323-338, Pergamon Infotech Limited, Maidenhead, Berkshire, England. 407 pages. The Rand Corporation.

Are Cartographic Expert Systems Possible?

Peter F. Fisher

Department of Geography, Kent State University,
Kent, Ohio 44242.

William A. Mackaness

School of Geography, Kingston Polytechnic,
Penryhn Road, Kingston Upon Thames, KT1 2EE, UK.

ABSTRACT

One of the major current thrust areas for computer software development is artificial intelligence and particularly expert systems. Several attempts have been made to implement cartographic design expert systems. None of these, however, can either understand why particular decisions are reached, or explain the reasoning to the user. This self-knowledge is one of the principle properties of any expert system and so it is doubtful whether any of the systems reported to date deserve the epithet "expert". This omission is not the fault of the system developers, but is caused by a lack of any systematised and accepted methodology for cartographic assessment. The cartographic community is urged to address this problem expeditiously.

INTRODUCTION

There can be no doubt that artificial intelligence and its associated programming techniques have made a major and increasing contribution to the field of computer science in recent years. A brief perusal of the shelves of any academic bookshop stocking computer science textbooks will reveal any number of tomes, too many to list here, with titles varying on the theme of Artificial Intelligence and related areas. Equally there is an increasing literature on the application of these programming techniques, particularly expert systems, to many science subjects. In the area of Geography and Geology, a recent review by the authors and others revealed eighteen expert systems of which details are published, while Waterman (1986) identifies some sixteen systems and any number of other systems are in preparation. These could all be described as experimental, to a degree, but have proved most successful and are in day-to-day use in the oil and mineral extraction industries (e.g. PROSPECTOR Cambell et al. 1982, MUD Kahn and McDermott 1984) and in environmental management (FIRES Davis et al. 1986).

In view of this it is not suprising to note that more and more expert systems are being proposed which proport to be applications of artificial intelligence techniques to cartography. It is the aim of this paper to briefly review these applications and assess the extent to which they can truly be described as expert systems.
For various reasons it is not possible to preview the contributions to this conference and so comments made here should not necessarily be taken to relate to those contributions.

ELEMENTS OF EXPERT SYSTEMS

In spite of the large number of published textbooks and papers which discuss expert systems there is no generally agreed definition of what constitutes an expert sytem. Waterman (1986, p.25), however, lists four essential properties of an expert system:

(1) Expertise which means having a high level of skill, exhibiting expert performance and having adequate robustness;

(2) Symbolic reasoning, which involves symbolic knowledge representation and the ability to reformulate symbolic knowledge;

(3) Depth, which is the ability to handle difficult problem domains and use complex tasks;

(4) Self-knowledge, which is examining its own reasoning and explaining its operation.

The symbolic knowledge is held in what is known as a <u>knowledge base</u>, the compilation of which requires a body of explicitly stated facts. These may be in the form of published literature or may have to be extracted from human experts in the domain for which the expert system is intended. Thus, in preparing a geological expert system, a geologist is consulted; for a cartographic system, a cartographer.

These facts are manipulated by the inference engine to derive diagnoses, prognoses, interpretations and other end products. Many ready-made inference engines are available at this point and provide a convenient route for the novice to concentrate on knowledge organisation without being concerned with programming.

At any particular point in the analysis, the expert system should have a <u>situation model</u> of the state of the "product". This can be reported to the user at any time for his approval. Similarly, the system should be able at any time to <u>justify</u> its line of reasoning or conclusion (Self-knowledge). Finally, expert systems can generally handle levels of uncertainty in the information supplied to them by the user.

A feature of expert systems which has caused some interest, but cannot be considered diagnostic, is the ability to update the knowledge base, according to further input by the user or expert. This so-called learning capability is in fact more complex than it may seem, although a simple example is given by Naylor (1983).

From the user's viewpoint, expert systems can be described as systems that for any particular set of input parameters can identify one particular outcome as the correct or the most probable from a large number of possible outcomes. In PROSPECTOR, this is achieved by Baysian combination of probabilities, so that essentially for any situation the most probably correct outcome is defined (Waterman 1986 p 55-57). This also enables the handling of uncertainty in user input.

CARTOGRAPHIC EXPERT SYSTEMS?

A number of cartographic expert systems have been described in the literature. These can be divided into systems for map and spatial data interpretation, systems to advise on how to produce maps; and systems for fully automated map production. These classes can be exemplified by the following systems: MAPSEE (Havens and Mackworth 1983), WERP (Taniguchi et al. 1984) and KBGIS (Smith et al 1984) are all involved to some extent with understanding maps and spatial data. The unnamed system presented by Muller et al (1986) supplies advice on how to construct maps, relating data types, among other variables, to map type, projection, etc; the consultation is purely verbal. Finally, the systems presented by Freeman and others (Freeman and Ahn 1984; Nicherson and Freeman 1986) and Pfefferkorn et al.(1985) all carry out some element of the map design process. The authors of this paper are principally interested in the design phase of cartography and so this paper is concerned primarily with the last of these groups; the reason for their apparent success is that they deal with just one element of map design, the objective of which is well-constrained and relatively easy to mathematically define. We are led to believe that such systems can be used as building blocks for more complex and "realistic" expert systems. This is not the case; the complete cartographic expert system will not be a set of independent design elements stacked together, but a system that considers all aspects of design at each stage of map compilation.

As noted above, one of the essential features of expert systems in other fields of application is that they are capable of taking a problem with a large number of possible outcomes and isolating the optimum solution. Similarly, the ability to justify the outcome is a pre-requisite of an expert system. Without wishing to detract from the quality of the software reported, neither of these requirements are met by the current map design expert systems. They all create one acceptable solution according to the production rules they are equipped with, without considering any other possible solutions. Name placement, the main area of cartographic production in which the authors are aware of expert system development (Freeman ansd Ahn 1984; Pfefferkorn et al. 1985) is an area where rules and evaluators (not overlapping previously positioned map names and features) can be clearly defined. There are some further areas of cartographic production where expert systems are under development, including parts of the generalisation problem (Nickerson and Freeman 1986). However, it is debatable whether any of these deserve the epithet expert system, principally because they have no self-knowledge, being wholly unable to explain their reasoning in establishing a particular outcome, or why one outcome and not another was achieved.

The task of knowledge engineering has been made still more difficult given that "no generally accepted concept of the process and functioning of cartographic communication exists" (Freitag 1980, p 18), an opinion shared by many other authors, if stated by them in other ways (eg Robinson 1975; Cuff and Matteson 1982). Self-knowledge for a cartographic expert system can only be achieved, however, when just such a concept exists and is accepted within the cartographic community. It is possible that the seminal work of Bertin (1984) may provide a basis for establishing such a concept, but there is much to be done to achieve it. Without clearly stated methods of assessment, a cartographic expert system can neither choose between alternative candidate designs, even in relatively simple areas of cartography such as name placement, nor act at all in more complex areas such as the assignment of symbols.

CONCLUSION

The authors are concerned that the development of cartographic expert systems will continue, irrespective of the lack of any widely accepted and clearly stated method of map assessment, and that if cartographers do not get on and develop the required methods, these expert systens will be produced by computer scientists not principally concerned with cartography.

In short, therefore, the authors firmly believe that cartographic expert systems are possible and, indeed, computer systems with some characteristics of expert systems are in existence. Self-knowledge, one of the major properties of an expert system is, however, at present entirely lacking and cartographers should be concerned to rectify this omission.

ACKNOWLEDGEMENT

The opinions expressed here were formulated during collaborative research with G Wilkinson, G Robinson and M Jackson, to whom we extend our appreciation.

REFERENCES

Bertin J, 1984 Semiology of Graphics: diagrams, networks, maps (translated from Semiologie Graphique), University of Winsconsin Press, Madison

Campbell A N, Hollister V F, Duda R O and Hart P E, 1982: Recognition of a hidden mineral deposit by an artificial intelligence program: Science, vol. 217, pp.927-929.

Cuff D J and Mattson M T, 1982, Thematic Maps: Their Design and Production, Methuen, London.

Davis J R, Hoare J R L and Nanninga P M, 1986, Developing a fire management expert system for Kakadu National Park, Australia: Journal of Environmental Management, Vol. 22, pp.215-217.

Freeman H and Ahn J, 1984, AUTONAP - An expert system for automatic name placement: Proceedings of the International Symposium on Spatial Data Handling, Aug 20-24, Zurich, Vol. 2, pp.544-569.

Freitag U, 1980, Can communication theory form the basis of a general theory of cartography: Nachrichten Aus Dem Karten und Vermessungswesen, Vol. 38, pp.17-35.

Havens W and Mackworth A, 1983, Representing knowledge of the visual world: Computer, Vol. 16, pp.90-96.

Kahn G and McDermott J, 1984, The MUD System; Proceedings of the first Conference on Artificial Intelligence Applications, IEEE Computer Society

Muller J-C, Johnson R D and Vanzella L R, 1986, A Knowledge-based approach for developing cartographic expertise: Proceedings of the Second Symposium on Spatial Data Handling, International Geographic Union, Williamsville, New York, pp 557-571

Naylor C, 1983, Build Your Own Expert System, Sigma Press, Wilmslow, UK

Nickerson B G and Freeman H 1986, Development of a rule-based system for automatic map generalisation: Proceedings of the Second International Symposium on Spatial Data Handling, Seattle, pp 537-556

Pfefferkorn C, Burr D, Harrison D, Heckman B, Oresky C and Rothermel J, 1985, ACES: A cartographic expert system: Proceedings of the Seventh International Symposium on Automated Cartography (AUTOCARTO 7), pp 399-407

Robinson A H, 1975, Map design, Proceedings of the Second International Symposium on Computer-Assisted Cartography (AUTOCARTO 2), pp 9-14

EXPERT SYSTEM INTERFACE
TO A GEOGRAPHIC INFORMATION SYSTEM

Bruce W. Morse
Senior Scientist
Autometric, Inc.
343 W. Drake Rd. #105
Ft. Collins, Colorado 80526
303-226-3282

ABSTRACT

Decisions concerning management of national forests require advice from experts of many different disciplines. Much of the information required from these forest management experts are spatial in nature; i.e. tree stand size, proximity to roads, slope, underlying soil type, etc. The analysis of this spatial data is facilitated using geographic information system (GIS). Yet the transfer of the expert's knowledge concerning spatial relationships related to forest management may require skilled use of a GIS, a skill which few experts in forestry possess. The system developed requires the forestry expert to provide the "if-then" relationship between the characteristics of the tree stand and the management recommendation. For instance, if an Aspen stand is 100 acres, on good soil, and within a quarter of a mile from a road, then that stand should be managed for timber, or whatever the expert decides. The "if-then" rules, which are easily entered into the expert system by the forestry expert, automatically access the GIS and its spatial database to provide forest management recommendations.

INTRODUCTION

Natural resource management has been radically altered through the introduction and use of automatedinformation systems. Resource managers are currently assisted by database management systems (DBMS) which handle their textual and numeric data and geographic information system (GIS) which handle their spatial data. More recentlyexpert systems are begining to be used to assist resource managers in handling their knowledge. Nevertheless the lack of true integration of these technologies can stifled the advantages these automated systems offer, such as speed and completeness in making management decisions. The lack of system integration has been recognized in handling text and numeric and spatial data resulting in the abilities of GIS to easily store and manipulate text and numeric data which has been associated with spatial data.

Analysis of text/numeric/spatial data is routinely accomplished through the interaction of experts familiar with the resource management decision at hand. The introduction and use of expert system technologies not only automates the storage and manipulate of knowledge, or expertize, it also affords the opportunity to intergrate the analysis of text/numeric/spatial data using the expertize embeded in an expert system.

This paper describes a system which interfaces an expert system with a GIS. The work performed for this effort was accomplished under a cooperative agreement between the U.S. Forest Service, University of Minnesota, and Autometric, Inc.

Problem Description

To properly understand the rational behind interfacing an expert system to a GIS, it is necessary to understand the type of work performed on the Nicolet. The Nicolet National Forest is located in Northern Wisconsin, where the predominate tree types are hardwoods and pines. The management of their aspen resources is of unique concern due to aspen's high value for timber production and wildlife (deer and grouse) habitat (Perala,1977). Deciding whether to manage an aspen stand for timber, wildlife, or even to manage it at all, requires knowledge contribute by a number of experts. These experts include professionals in the fields of silviculture, wildlife management, pest management, and soil science. Traditionally these experts evaluate the site characteristics of an aspen stand and recommend a particular management action based on their intimate knowledge regarding the relationships between site characteristics and subsequent stand production. For instance, a small aspen stand containing young trees growing on average soils far from an access road should be managed for wildlife, while a much larger stand should be managed for timber. These types of relationships can be formulated into "if-then" rules, which can be incorporated into a rulebase residing in an expert system.

The vast majority of aspen management rules make use of relationships which are spatial in nature, such as stand size, spatial distribution, proximately to roads, underlying soil types, etc. This spatial information can be obtained from conventional analysis techniques using geographic information systems. Currently the Nicolet National Forest is using the public domain GIS, known as MOSS, to perform such spatial calculations and analysis (Anonymous, 1984).

ASPENEX automates the analysis of site characteristics by intergrating the knowledge in an expert system with the analytics of MOSS. The rules, used in the expert system, were developed with the assistance from aspen management experts, who work on the Nicolet. MOSS was modified to accept very specific data exchanges between it and EXSYS. Communication software was developed to pass data and instructions between the personal computer and Data General. Initiation of the system occurred on the personal computer by running a program which controls and orchestrates the activation of the communication software. This in turn activates MOSS programs on the D.G. where spatial analysis are performed and creation of a file containing the necessary aspen site characteristics. This file is transferred back to the personal computer were it is analysised by expert system's rulebase. A final report is generated prioritizing the recommended management options based on the site characteristics. This file is used by the forest manager to assist in the final management actions for the aspen stands.

SYSTEM DESCRIPTION

Hardware Configuration

The Nicolet National Forest operates a Data General (DG) MV/8000 in the Forest Supervisor's office. The District Offices have the capability to network off this computer. The DG computer was originally procured for office automatic functions, thus other application software operation on the DG are not encouraged. Yet the access to personnel computer are nevertheless becoming more commonplace, which is helping to distribute to usage of central computer processing time. The development of ASPENEX was designed to take advantage of this current hardware situation. ASPENEX performs the bulk of its calculations on a personal computer, accessing the DG only when necessary.

Software

ASPENEX is composed of four major software components; Geographic Information System, Expert System, Communication, and Control. Each component is described in detail in the following sections.

Geographic Information System:

In 1982, the public domain Geographic Information System known as MOSS, was installed on the Nicolet National Forest's DG. Since then over 20 different themes of spatial information have been digitized for spatial analysis using MOSS. These themes comprise the spatial information required for everyday forest management and long-term forest planning. The themes in the MOSS database includes, but are not limited to, timber stand maps, soil types, transportation, hydrology, wildlife and pest information, and land ownership. It is estimated that this MOSS database requires over 100 megabytes of disk storage. Currently the Nicolet uses of MOSS include pest management, grouse habitat identification, forest planning, etc. (Anonymous,1984).

ASPENEX takes advantage of both the digital spatial database and the spatial analysis capabilities offered by MOSS. The MOSS database is encoded so that each map feature (i.e. tree stand) is uniquely identified by its primary attribute or subject (i.e. stand type). The feature subject can be used to sort and select map features containing the search criteria. Further analysis of the data can be performed using MOSS functions, such as area calculations, proximity determinations, etc. MOSS analysis functions are modular, so that very specific information can be obtained from the database. For example MOSS can identify all mature aspen stands larger than 40 acres which are growing on good soil, yet are within an half a mile from an access road. ASPENEX makes use of these types of retrievals.

Expert System. In the last few years a number of expert system shells have become commercially available for expert system development and applicabtion. ASPENEX uses one such shell known as EXSYS. EXSYS was selected because it requires minimal trainning, it operates on most personal computers, and it allows for inter-program communication, such as with MOSS.

Communication. The existing hardware/software configuration placed the GIS and data on the Data General and the expert system on the personnel computer. In order for ASPENEX to automatically query and transfer data between MOSS and EXSYS, communication software was developed. Communication software handled all inter-machine transfer of data, as well as initiation of program execution on the different computers.

Program Control. The execution and control of the various components of ASPENEX were orchestrated by a control program running on the personnel computer. This program performed a number of different functions to assure proper operation of ASPENEX. These functions included the following:

. User interface
. Formating of data
. Execution of ASPENEX components
. Error checking

SYSTEM OPERATION

Rule-Base Creation

A primay consideration for selecting EXSYS was its ease in its operation of the rule-base creation, both of which are well documented (EXSYS, 1985). The building of the rule-base, the knowledge-engineering process, involved searches of available public knowledge and extraction of the private knowledge from recognized experts in management of aspen resources (Graklandoff, 1985).

The final recommendation for aspen management can be one of three actions. These actions are: manage the stand for wildlife, manage the stand for timber, or do not manage the stand at all. Deciding which action to take, requires information on the tree stand, as well as nearby associations. Typically the site characteristics include, stand age and density, stand size and species type, underlying soil, distance to access roads, and spatial distribution of related stands. The management action for each stand is judged separately based on the stand's site characteristics. The relationship between action and site characteristics is determined from the knowledge-engineering process. Each variable (site characteristic) has associated with it a relationship for each management action. From this relationship, ASPENEX calculates a value which indicates the possibility, between 0 and 100, for recommending that action. These relationships were determined based on the knowledge found in published sources (Basham 1958, Perala 1977, Shields etal. 1981, and Walters 1982), as well as from personnel interviews with Aspen experts. Depending on all the site characteristics, the final possibility will determine the final recommendation. The final possibility is calculated from the average of all individual possibilities.

Spatial Analysis

The method in which ASPENEX determines the site characteristics is through the execution of MOSS functions using the Nicolet spatial database. As mentioned in the previous section, information pertaining to, stand type, stand size, soil type, and distance to roads were all necessary before recommending a management action. This information is spatial in nature, thus can be derived from the MOSS geographic information system.

ASPENEX Output

Final output from ASPENX is in the form of a report summarizing stand characteristics derived from MOSS and final composit score calculated by EXSYS using the stand information (Table 1). Each stand has associated with it three scores, between 0 and 100, which correspond to the management action, timber production, wildlife habitate, or ignore. The type of management favored is based on the highest score from EXSYS. The composit score is calculated using the average of all the individual values when the rules were found to be true.

The output report can be further evaluated and summarized in database management system a statistical package. The data can also be transfered back into MOSS for creation of a map identifying and locating those stands recommended for a particular management action.

****

Table 1. Example of report generated ASPENEX. "Possibility of Management" values calculated by expert system using spatial information provided by geographic information system.

****

Stand ID*	Area	Soil Code	Prox. to Road
02219016913	35.35	2	No
02221003913	24.75	2	Yes
02220002122	121.44	5	Yes

	Possibility of Management		
Stand ID*	Timber	Wildlife	Ignore
02219016913	64	82	35
02221003913	71	73	21
02220002122	64	57	30

*Stand ID contains information on type and size of trees within stand.

****

SUMMARY

ASPENEX is a prototype system developed to assist in the forest management of aspen on the Nicolet Naitonal Forest. ASPENEX intergrates`an expert system with a geographic information system. The expert system provides the rules required to manage aspen, while the geographic information system provides spatial information on the characteristics of the aspen stand. The spatial site characteristics are automatically passed to the expert system, where the data is analyzed using the rulebase created by the aspen experts. ASPENEX is currently being used by the Nicolet National Forest and enhancements are being made to the system for application to other management concerns.

ACKNOWLEDGMENTS

The author to thank William White (USFS), Laura Umland (Nicolet National Forest), John Davidson (Autometric, Inc.), and Bill Turner (Autometric, Inc.) for their valuable support. This work was supported through a cooperative project between the U.S. Forest Service, University of Minnesota, and Autometric, Inc.

REFERENCES

Anonymous, 1984. Use of Map Overlay and Statistical System on the Nicolet National Forest. Unpublished report on file at Nicolet National Forest, Rhirelander, WS. pp.13.

Burrogh, P.A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Clarendon Press, Oxford. pp. 193.
Basham, J.T. 1958. Decay of Trembling Aspen, Can. J. Botany, 36:491-505.

EXSYS, Inc. 1985, EXSYS Expert System Development Package. Albuquerque, NM. pp.86.

Graklandoff, G.A. 1985. Expert System technology Applies to Contographic Processes-- Considerations and Possibilities. Presented at Fall 1985 ASPRS Conference-Indianapolis, IN. , pp. 613-624.

Perala, D.A. 1977. Manager's Handbook for Aspen in the North Central States. USDA Forest Service, N.C. Forest Exper. Stat. General Tech. Report NC-36. pp. 30.

Shields, W.J. and J.G. Bockheim. 1981. Deterioration of Trembling Aspen Clones in the Great Lakes Region. Can. J. For. Res. 11:530-7.

Walters, J.W. 1982. Evaluation of a System for Predicting the Amount of White Trunk Rot (Phellinus Tremulae) in Aspen Stands. Unpublished report on file at U.S. Forest Service, Forest Pest Management St. Paul, MN Field office. pp.7.

AUTOMOBILE NAVIGATION IN THE PAST, PRESENT, AND FUTURE

Robert L. French
Consultant
4425 W. Vickery, Suite 100
Fort Worth, Texas 76107

ABSTRACT

One of the first automobile navigation systems appeared around 1910 in the form of route instructions printed on a turntable-mounted disk driven by an odometer mechanism in synchronization with distance travelled along the route. Instructions keyed to specific distances from the beginning of the route came into view under a pointer at the time for execution. Proximity beacon navigation, first researched in the United States during the 1960s, has largely given way to autonomous map-matching systems and to advanced radio-location systems in the 1980s. Autonomous systems achieve high relative accuracy by matching observed mathematical features of dead-reckoned vehicle paths with those of road networks encoded in a map data base, but occasionally require manual resetting to a known location. Satellite-based navigation systems offer high absolute accuracy, but require dead-reckoning augmentation because of signal aberrations in the automotive environment. Thus future systems are likely to incorporate multiple navigation technologies. The main developments yet to come are in the information and institutional areas. Private sector investments will be required for the development and maintenance of comprehensive digital map data bases, and coordination among public sector organizations will be required for collecting, standardizing, and communicating real time information on traffic and road conditions.

INTRODUCTION

Industry leaders are beginning to take it for granted that sophisticated navigation systems for automobiles will become commonplace by the end of the 1990s (Rivard 1986). The stage is being set by high-technology systems developed and tested (and, in some cases, marketed) during the 1980s. But few are aware of the surprisingly rich history of what had been accomplished in vehicular navigation long before the present decade. In fact, as we enter an era of high technology automobile navigation, we find relatively little underlying technology that is basically new, other than the on-board digital computer. The computer enables advanced radio-location schemes, and it makes map matching possible, thus breathing new life into dead-reckoning technologies that are ancient compared to the automobile itself. This paper describes the past, present and future of automobile navigation in terms of developments prior to 1980, those of the present decade, and those that may be expected beyond 1990.

AUTOMOBILE NAVIGATION IN THE PAST

Early developments relating to vehicle navigation technology are listed in Table 1. Virtually all high-technology automobile navigation systems use on-board computers to integrate and automate two or more of these technologies to provide vehicle location, heading, routing, or step-by-step route guidance. A historical overview of automobile navigation technology is given in an earlier paper (French 1986). Highlights are summarized below.

Table I. Vehicle Navigation Milestones

DATE	TECHNOLOGY
<60 AD	Odometer
200-300	Differential Odometer
1100-1200	Magnetic Compass
1906	Gyrocompass
1910	Programmed Routes
1940	Loran Positioning
1964	Satellite Positioning
1966	Proximity Beacon
1971	Map Matching

South Pointing Carriage

The South Pointing Carriage is the earliest known example of a land vehicle navigation system. This direction-keeping device is a Chinese invention dating back to 200 - 300 A.D., possibly earlier. Chinese literature confused the south-pointing carriage with the magnetic compass (invented almost 1000 years later) so thoroughly that historical research has only recently established that the south-pointing carriage had nothing to do with magnetism. Instead, it was based on the principle (now called "the differential odometer") that for a given change in vehicle heading, a vehicle's outer wheels travel a mathematically-predictable distance farther than the inner wheels. When changing heading, a gear train driven by a south-pointing carriage's outer wheel automatically engaged and rotated a horizontal turntable to exactly offset the change in heading. Thus a figure with an outstretched arm mounted on the turntable always pointed in the original direction regardless of which way the carriage turned.

Jones Live-Map

Among the first U. S. devices for car navigation was the Jones Live-Map introduced in 1909. This mechanical road guide consisted of a turntable driven by a gear train connected by flexible shaft to one of the vehicle wheels. Paper discs for individual routes had a scale of miles printed around their perimeter and were mounted on the turntable beneath a glass cover with a fixed pointer. Printed road directions keyed to specific distances from the beginning of a route came into view under the pointer at the time for execution. An advertisement for the Jones Live-Map claimed "You take all the puzzling corners and forks with never a pause. You never stop to inquire ...."

Chadwick Road Guide

The Chadwick Road Guide, another odometer-driven device introduced in 1910, had signal arms and a bell activated by punched holes in a programmed route disc. As each maneuver point was approached, one of ten signal arms bearing color-coded symbols indicating the action to be taken appeared behind a window and the bell sounded to attract the driver's attention. A Chadwick advertisement read:

> "The Chadwick Automatic Road Guide is a dashboard instrument which will guide you over any highway to your destination, instructing you where to turn and which direction. You will be warned upon your approach to rough roads, railroad tracks, speed traps. The names of the city or town through which you are passing and the name of the street will appear on your Chadwick Road Guide. Model B - $55.00, Model C - $75.00."

Short of automatically maintaining synchronized position, the Chadwick Road Guide is strikingly similar to modern concepts for real-time route guidance.

Vehicular Odograph

The vehicular odograph, a self-contained navigation system for jeeps and other U. S. Army vehicles, was developed during WWII (Faustman 1945). An electromechanical system drove a stylus to automatically plot vehicle course on a map of corresponding scale. An odometer provided a distance input measurement which was mechanically resolved into x,y components using servo-driven input from a photo-electrically-read magnetic compass. The vehicular odograph, the first example of an automated system for determining and showing vehicle location on a map, is a precursor of state-of-the-art systems for CRT display of vehicle location on a digital map. A post-WWII publication (McNish and Tuckerman 1947) speculated about the potential for civilian automobile use of the vehicular odograph:

> "One is inclined to wonder if cost would prove an important limitation if a cheaper model of the odograph were manufactured by the millions, and maps of cities and of tourist routes, drawn to the proper scale for use with the odograph, were available at every filling station."

Driver Aided Information and Routing (DAIR) System

Proximity beacon navigation, which uses strategically positioned short-range location-coded signals, was first researched in the United States starting with DAIR in the mid-1960's. DAIR, which used roadbed arrays of magnets arranged in binary code to communicate location to passing vehicles and was the subject of limited development and testing by General Motors, was a forerunner of ERGS.

The Electronic Route Guidance System (ERGS)

ERGS, which was researched by the Federal Highway Administration during the late 1960's as a means of controlling and distributing the flow of traffic (Rosen, et al. 1970), is based upon automatic radio communication with roadside equipment to provide equipped vehicles with individual routing instructions at decision points in the road network. An in-vehicle console with thumbwheel switches permits the driver to enter a selected destination code. The code is transmitted when triggered by a roadside unit as approaching key intersections. The roadside unit immediately analyzes routing to the destination and transmits instructions for display on the vehicle console panel. Although technically sound, ERGS required expensive roadside infrastructure and the development effort was terminated by Congressional mandate in 1970.

Automatic Route Control System (ARCS)

Networks of roads and streets may be modeled as internodal vectors in a digital map data base, and a particular route may be "programmed" as a unique sequence of mathematical vectors. As demonstrated in 1971 (French and Lang 1973), an on-board computer may be programmed to analyze dead-reckoning inputs and match the deduced vehicle path with programmed routes to automatically remove position discrepancies that would otherwise build up. The automatic route control system (ARCS) used a differential odometer for dead reckoning and a map-matching algorithm to correlate each sequentially measured vector with its data base counterpart. The vehicle's location along the route was confirmed, and pre-recorded audio route guidance instructions were issued where appropriate. A second version issued visual route instructions on a plasma display panel (French 1974). ARCS yielded an average location accuracy of 1.15 meters during extensive tests over newspaper delivery routes which were originally "mapped" by driving an ARCS-equipped vehicle over them while operating in a data acquisition mode.

STATE-OF-THE-ART SYSTEMS

A variety of automobile navigation systems have appeared during the 1980s. Most state-of-the-art systems fall within the following classifications:

Dead Reckoning

A dead-reckoning system called "City Pilot" is now on the European market. Developed by VDO Adolf Schindling AG, it uses an earth magnetic field sensor and an odometer distance sensor (Gosch 1986). Prior to a journey, the driver uses a light pen to read bar-coded starting and destination coordinates on a special map. Using the sensor inputs and destination coordinates, a microcomputer calculates the direction and line-of-site distance to the destination. LCD arrows show the driver which general direction to take, while numerals indicate the distance. Test results reveal that drivers using the system reach their destinations with an accuracy of 97 percent (i.e., within 3 percent of the distance travelled).

Other recent examples of dead-reckoning systems include the Nissan "Driver Guide", the Honda "Electro Gyro-Cator", and the Daimler-Benz "Routenrechner". The Nissan system (Mitamura , et al. 1983) uses magnetic compass and odometer signals to continuously compute the distance and direction to a destination whose coordinates are input by the driver. A display comprised of an array of symbolic indicator lights shows the current direction to the destination, and a bar graph shows remaining distance.

The Honda system (Tagami, et al. 1983) uses a helium gas-rate gyro and odometer to compute the vehicle's path relative to its starting point. The path is displayed on a CRT screen behind a transparent map overlay of appropriate scale. Provision is included for manually adjusting the map position to keep it in registration with vehicle path.

The Daimler-Benz system has two modes, one for city and one for highway driving (Haeussermann 1984). The city mode operates much like the Nissan system, using magnetic compass and odometer inputs to compute and display distance and direction to a driver-specified destination. In the highway mode, the system makes use of stored digital map data for the highway network. The driver inputs origin and destination, and the system computes the optimum highway route and prompts the driver step-by-step over the route. The next route point and its distance is continuously shown on a small alphanumeric display. Only odometer input is used in the highway mode; the driver must manually correct any distance error when arriving at route points.

Proximity Beacon

This approach, now inactive in the U. S., has been the subject of further development and testing in Japan and West Germany. The major new development in proximity systems is ALI-SCOUT, a joint project of the West German Government, Siemens, Volkswagen, Blaupunkt and others (von Tomkewitsch 1986). ALI-SCOUT is a route guidance system that receives area road network data and recommended route data from strategically-located IR beacons. Simplified graphic driving directions to the input destination are presented in real time on a dashboard LCD. Destination input, as well as system control, is via a hand-held wireless remote-control unit. ALI-SCOUT will be subjected to large-scale user tests in West Berlin starting this year.

Map Matching

The first commercially available automobile navigation system based on map-matching technology is the Etak Navigator™ now marketed in California. The Etak system uses a flux-gate magnetic compass as well as differential odometry and uses 3.5-MByte tape cassettes to store digital-map data (Honey and Zavoli 1985). The vehicle's location relative to its surroundings is continuously displayed on a CRT map which may be zoomed to different scales. A fixed symbol below the center of the CRT represents the vehicle position, and points to the top of the display indicating vehicle heading. As the vehicle is driven, the map rotates and shifts about the vehicle symbol accordingly. Input destinations are also shown on the Etak screen.

A map-matching system developed for testing and demonstration by Philips features the compact disc (CD-ROM) for storage of map data bases (Thoone and Breukers 1984). Called "CARIN", this system includes a route-search algorithm and provides step-by-step route guidance. A color CRT map display shows vehicle location relative to the surroundings, and voice instructions prompt the driver when operating in the route guidance mode.

Bosch-Blaupunkt has developed a map-matching system called "EVA" which uses a differential odometer and includes route-search software to generate explicit route-guidance instructions (Pilsak 1986). Turns at intersections, lane changes, etc. are specified on an LCD in the form of simplified diagrams which show lane boundaries and use arrows to indicate the path to be taken. Voice capability is included, and is used to confirm destination entries. A CD-ROM version is under development.

Satellite

The Transit navigation system, implemented by the U. S. Navy and operational since 1964, was the basis for a Ford concept car navigation system (Gable 1984). Several Transit satellites in polar orbits at a height of approximately 1,075 kilometers are longitudinally-spaced to give worldwide, albeit intermittent, coverage. Each satellite transmits information which, in combination with measured Doppler characteristics, permits calculation of receiver location by iterative solution of a set of equations. Since a Transit satellite is not always in range, the Ford system included dead reckoning for continuous determination of position between satellite passes. The vehicle speedometer and a flux-gate magnetic compass software-compensated for magnetic variations provided dead-reckoning inputs. A touch-screen color CRT provided alternative displays, including vehicle heading in several formats and a map display with cursor tracking of vehicle position.

The Navstar Global Positioning System (GPS), which is being implemented by the Department of Defense, has been considered as a basis for automobile navigation systems by both Ford and General Motors (Gable 1984), and was the basis for CLASS, the Chrysler Laser Atlas and Satellite System, a concept displayed at the 1984 World's Fair in New Orleans (Lemonick 1984). CLASS included a nationwide set of maps stored in image form on a video disc, and software for automatically selecting and displaying on a color CRT the map area incorporating the vehicle's current location as indicated by a cursor.

Still in the implementation stage, the Navstar GPS system will be completed in the early 1990s when the last of 18 satellites are orbited. The 18 satellites are being spaced in 12-hour orbits such that at least four will always be in range from any point on earth. Using timed signals from four satellites, the receiver's computer automatically solves a system of four simultaneous equations for its three position coordinates and a time bias signal for synchronizing the receiver's quartz clock with the satellites' precise atomic clocks. The signals are modulated with two pseudo-random noise codes: P, which provides position accuracies as close as 10 meters, and C/A which is about one tenth as precise. Use of the P code may be restricted to authorized applications.

Although GPS has great potential accuracy and will provide continuous coverage, auxiliary dead reckoning is required in automobile applications to compensate for signal aberrations due to shadowing by buildings, bridges, foliage, etc. A recent evaluation of GPS for land vehicles notes that, because of differing ellipsoidal reference systems, the task of melding GPS location with local maps is formidable (Mooney 1986). Hence map matching technologies may be useful with GPS as well as with dead reckoning.

AUTOMOBILE NAVIGATION IN THE FUTURE

Figure 1 shows elements and functions likely to appear in automobile navigation and information systems during the balance of this century. Dead reckoning will be included even though precise location sensing, such as Navstar GPS, will probably be available at acceptable cost. Distance and heading sensing may be accomplished by differential odometry using input signals from anti-lock braking systems, or in combination with software compensated fluxgate magnetic compasses. The fiber-optics gyroscope also shows potential as an inexpensive and rugged means for accurately sensing heading changes.

Map matching, an artificial intelligence process, will have a role in many future systems. Map matching based upon dead reckoning alone occasionally fails if digital maps are not current, or from extensive driving off the defined road network. Thus absolute location means, such as satellite positioning or proximity beacons, will be required if manual reinitialization is to be avoided. Although it appears unlikely that the proximity-beacon approach will be pursued again in the United States in the foreseeable future, proximity beacons may provide additional navigation inputs in some countries.

Large capacity storage, such as the CD-ROM, is required for useful amounts of map data. A nationwide digital map would fit on one 500-MByte CD-ROM disc. However, future systems will also allocate large amounts of storage to "yellow pages" or other directory information to aid drivers in performing personal errands and business functions.

Data communications will be a feature of future systems in countries that integrate traffic management with in-vehicle route guidance to enhance the benefits. One-way communications from the infrastructure to the vehicle is the most useful link. However, additional benefits would be provided by vehicle-to-infrastructure data communications. This could provide destination information to central traffic management systems for planning optimal traffic flow or, as in the case of the ALI-SCOUT (von Tomkewitsch 1986), eliminate the need for traffic sensors by reporting recent travel experience to the central traffic management system.

Driver inputs and system outputs are provided through a driver interface, an important and controversial system element which must take into account ergonomics and safety considerations as well as functional requirements. Most U. S. systems proposed or developed to date use detailed

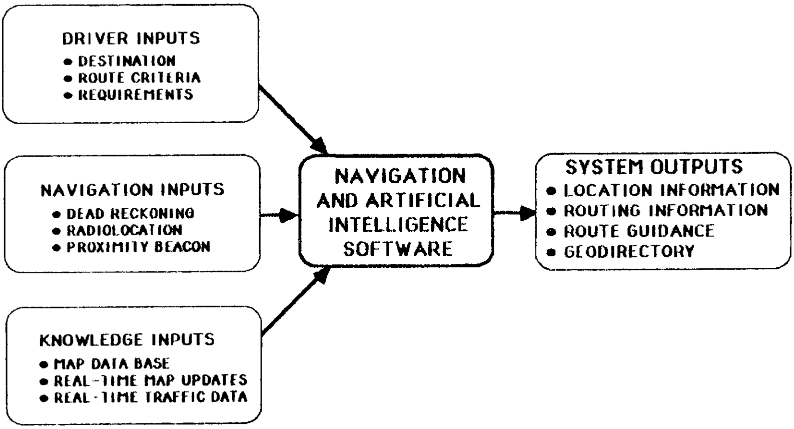


Figure 1. Future Navigation System

map displays, whereas most European systems use some combination of symbolic graphics, alphanumeric messages, and audio signals. Future systems will probably have multiple modes of operation and some will have "heads up" displays for route guidance. Driver inout, presently by key pad or touch screen means, will eventually include voice recognition.

Long-term scenarios for car navigation and information system development and deployment are projected by ARISE (Automobile Road Information System Evaluation), a 1985 feasibility study performed by the Swedish consulting firm ConNova:

1990 - Autonomous, vehicle-born navigation systems are available. The main market is for commercial vehicles.

1995 - Navigation devices in commercial vehicles (e.g., trucks, taxis and limousines) have become commonplace.

2000 - New integrated road information systems give route and navigation assistance, and road traffic is flowing more smoothly.

2010 - Integrated road information systems are now fitted in half of all road vehicles, and is becoming standard equipment in new cars.

CONCLUSIONS

Car navigation based upon dead reckoning alone is limited in both accuracy and application. These limitations may be alleviated by map matching, but map matching requires digital maps. Advanced radio-location technology has good potential for high absolute accuracy, but requires dead-reckoning augmentation. Proximity beacons for navigation also provide good accuracy but require extensive roadside infrastructure. Interactive car navigation and route guidance requires integration with traffic management systems

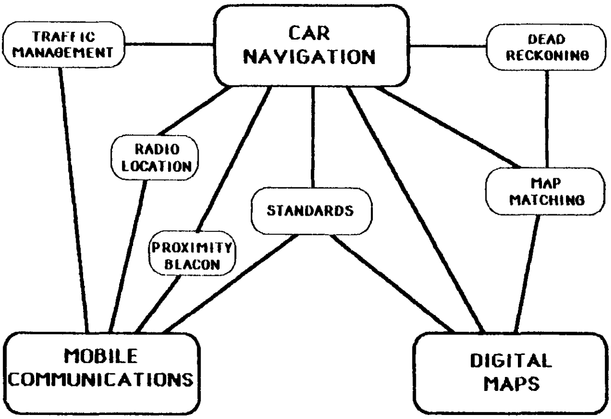


Figure 2. Navigation, Maps and Mobile Communications

which, in turn, requires data communications. Finally, car navigation, digital maps, and mobile data communications all require effective standards. Figure 2 illustrates some of the more important interrelationships.

Practical technologies for future automobile navigation and route-guidance systems have already been developed, or are within reach. Digital maps, vital to advanced automobile navigation and information systems, are becoming available, but early versions define little more than basic road geometry, street names, address ranges, and road classification. Maps for in-vehicle systems that compute best routes and provide automatic route guidance also require traffic regulation and other attributes that may influence route choice. Systems developers and map firms have been reluctant to undertake comprehensive digital map development before the market becomes better defined. However, standardization efforts by the new SAE Automotive Navigational Aids Subcommittee provide encouragement.

A remaining obstacle to advanced car navigation systems in the United States is the lack of established means for collecting and communicating real-time traffic data, as well as updated map data, for use in dynamic routing. Although mobile data communications will be essential to advanced car navigation and route-guidance systems, the U.S. public sector, unlike other developed countries, has no coordinated effort to bring about the necessary standardization.

REFERENCES

Faustman, D. J., 1945, New Mapping Instrument by Corps of Engineers: Surveying and Mapping, Vol. V, No. 3, 30-37.

French, R. L., G. M. Lang, 1973, Automatic Route Control System: IEEE Transactions on Vehicular Technology, Vol. VT-22, 36-41.

French, R. L., 1986, Historical Overview of Automobile Navigation Technology: Proceedings of the 36th IEEE

Vehicular Technology Conference, 350-358, Dallas, Texas.

French, R. L., 1974, On-Board Vehicle Route Instructions via Plasma Display Panel: SID International Symposium Digest of Technical Papers, Vol. 5, 146-147.

Gable, D., 1984, Automobile Navigation: Science Fiction Moves Closer to Reality: Electronic Engineering Times, Vol. 296, D14, D18.

Gosch, J., 1986, Smart Compass Pilots a Car to Its Destination: Electronics, Vol. 59, No. 21, 20-21.

Haeussermann, P., 1984, On-Board Computer System for Navigation, Orientation and Route Optimization: SAE Technical Paper Series, No. 840485.

Honey, S. K., W. B. Zavoli, 1985, A Novel Approach to Automotive Navigation and Map Display: RIN Conference Proceedings - Land Navigation and Location for Mobile Applications, York, England.

Lemonick, M., 1984, Now: Driving by Satellite: Science Digest, Vol 92, 34.

McNish, A. G., B. Tuckerman, 1947, The Vehicular Odograph: Terrestial Magnetism and Atmospheric Electricity, Vol. 52, No. 1, 39-65.

Mitamura, K., S. Chujo, and T. Senoo, 1983, The Driver Guide System: SAE Technical Paper Series, No. 830660.

Mooney, F. W., 1986, Terrestrial Evaluation of the GPS Standard Positioning Service: Navigation, Vol. 32, 351-369.

Pilsak, O., 1986, EVA - An Electronic Traffic Pilot for Motorists: SAE Technical Paper Series, No. 860346.

Rivard, J. G., 1986, Automotive Electronics in the year 2000: Proceedings of the International Congress on Transportation Electronics, 1-18, Dearborn, Michigan.

Rosen, D. A., F. J. Mammano and R. Favout, 1970, An Electronic Route Guidance System for Highway Vehicles: IEEE Transactions on Vehicular Technology, Vol. VT-19, 143-152.

Tagami, K., T. Takahashi, and F. Takahashi, 1983, Electro Gyro-Cator, New Inertial Navigation System for Use in Automobiles: SAE Technical Paper Series, No. 830659.

Thoone, M. L. G., R. M. A. M. Breukers, 1984, Application of the Compact Disc in Car Information and Navigation Systems: SAE Technicl Paper Series, No. 840156.

von Tomkewitsch, R., 1986, ALI-SCOUT - A Universal Guidance and Information System for Road Traffic: Proceedings of the IEE International Conference on Road Traffic Control, 22-25, London, England.

Digital Map Requirements of Vehicle Navigation

Marvin White
Etak, Inc.
1455 Adams Dr.
Menlo Park, CA 94025
(415) 328-3825

ABSTRACT

We consider digital maps for electronic navigation systems, and present characteristics of the map and its presentation that are required for navigation assistance. The Etak Navigator™ and its requirements serve as examples. The most important are topological structure, positional accuracy, identifying information like names and addresses, and street classification. Topological structure is needed for retrieving data in the ever-changing neighborhood of a moving vehicle. Positional accuracy is needed in navigation systems that, like Etak's, use map-matching. Identifying information is needed both for display and destination finding. Street classification is used in several ways: to display different classes of features differently, to select data to show at various scales, and in routing. We also consider the merits of various forms of presentation and their effects on the map requirements. For example, a visual heading-up display has the merit of giving an oriented context to the local navigational information.

INTRODUCTION

Automated navigation aids impose varying requirements on the supporting digital map. With sufficient information in the database it is possible to present a new map on a screen or compose new or updated instructions at each significant change in the driver's situation. Such changes include turning onto a different road, advancing along the same road, and changing one's mind. It is also possible to abstract essential navigational data for particular situations and drop unimportant detail. These can be powerful aids to a driver.

Navigation always involves knowing your current location, finding your destination, usually with respect to a map, retrieving relevant maps or map data, and presenting the data to the driver. The Etak Navigator™ constantly tracks current location by dead reckoning augmented with map matching, i.e. matching the dead-reckoned path to possible paths on the road map (Zavoli 1986). This imposes topological and metrical requirements on the digital map content and speed requirements on retrieval.

Destination finding can impose heavy demands on the digital map. To find street addresses or intersections, for example, requires a great deal of data as well as high speed retrieval.

Map retrieval is also crucial to navigation assistance and deserves consideration apart from map content. It is a support function at the core of navigation assistance and related applications.

Presentation of the data to the driver is also always a part of navigation assistance. Presentation can be graphical with simple interaction, as in the Etak Navigator™ or at the other extreme can be driving instructions written or given by voice. There are many differences beyond human factors considerations; in particular, digital map support for graphics is more demanding in coordinate accuracy and for instructions is more demanding in traffic flow restriction data, such as one-way or turn prohibitions.

We consider below the implications of providing various kinds of navigation assistance on digital map content, retrieval and how these relate to presentation. Neukirchner and Zechnall have presented their findings on map requirements with particular emphasis on map sources and precision (Neukirchner 1986). In this paper, we emphasize categories of information required in relation to retrieval and presentation. We find:

- topology to be fundamental to both content and retrieval;
- metrical data (coordinates) important for map matching and display;
- names, addresses and geographic codes required for destination finding;
- cartographic generalization important for both display and routing; and
- many kinds of ancillary data useful in many ways.

Fast retrieval from a huge store is needed to keep track of current location, find destinations, and find paths in the network. We also consider interactive graphic presentation and its demands on retrieval and content in contrast to giving instruction and its greater demands on completeness and content.

CONTENT

The simplest aspect of a digital map to consider is content. Format and organization are related and are sometimes called data structure. It is important to understand that the structure, if you have topology in mind, is the data, or at least part of the data content and must be captured just as street name or priority must.

Topology

There have been demonstrations of navigation systems that use images of maps and entirely avoid encoding topology. The Etak operates at at the opposite extreme: we use topology nearly everywhere.

Even if the only use were in destination finding (say finding intersections), while navigation, display, etc. depended not at all on topology, it would be worthwhile to store the map as a topological database. In the first place, capturing the intersections is the bulk of the topological work and in the second, topological map storage is much more efficient than image storage.

Additional reasons for using a topological database abound. Finding a path through a network is a topological calculation. Connectedness (a topological property) is useful in map matching. As a vehicle advances on the ground, so does its relevant neighborhood; retrieval of a new neighborhood related to the last is a topological operation. Annotation and ancillary data is efficiently associated with elementary topological data; this is important for both destination finding and presentation of data.

Taken together, these facts are compelling evidence that to provide navigational assistance of even modest sophistication topological data must be included and further the database must be organized topologically to permit topological operations.

Metrical data

Coordinates are often regarded as the quintessential map data and a great deal of attention is given to coordinate reference systems with passionate pleas for particular ones, such as UTM. This is a mistake. Topology is more important for the reasons mentioned above as well as deeper mathematical and philosophical reasons (Corbett 1979). It is possible to give effective instructions for driving from one point to another using no metrical data and one can even imagine diagrammatic graphic presentations using no metrical data. So coordinates are unnecessary for some very useful navigation aids.

Nevertheless, distances and headings are helpful in driving instructions. Coordinates are very useful in graphic presentations and required for map matching or interpreting radio navigation data. So coordinates are important in navigational assistance.

Coordinate accuracy requirements vary with the application. Etak's map matching algorithms, for example, impose accuracy requirements on the coordinate data equal to the requirements for USGS 1:24000 quads. these requirements are local; relative accuracy is critical global accuracy hardly matters. One reference ellipsoid is as good as another and, as is the case with most extant maps, different ellipsoids for different continents work well. Because of their applications to geodesy, coordinate systems used in topographic map series are usually geodetic using a reference ellipsoid whose axis does not coincide with the Earth's but is parallel.

Etak's Navigator also dynamically recalibrates the sensors by gleaning information from the map matching. This means that position errors in the digital map not only cause failure to match and incorrect matches, but also degrade calibration, making navigation performance worse. So relative coordinate accuracy is very important for Etak's navigation both in map matching and in maintaining navigation performance in the dynamic environment of a car.

Radio navigation requires coordinates matched to the particular system, be it LORAN, Geostar or GPS. Relative accuracy is less important here; indeed local distortions matching characteristics of the particular system may be helpful. Satellite systems, being global in nature, use a single geocentric reference ellipsoid rather than the set of ellipsoids used for map series in different continents.

The high precision promised by GPS implies that GPS coordinates will meet the relative accuracy requirements of the Etak Navigator™. A GPS receiver could be connected to a Navigator as an additional sensor that works especially well in open terrain. This would be adding an absolute navigation device to a dead-reckoning device, which is by its nature relative.

Using GPS or other absolute navigation devices puts additional requirements on the map. The coordinates output by the GPS receiver must be transformed to those of the map to be displayed (or visa versa) so that position relative to objects on the ground is known, which is essential for navigation. In fact, it is far more important to know your position relative to nearby objects than to a set of satellites. The ability to accept data gathered using different techniques and interpreted using different coordinate systems is and will remain important. It won't suffice to declare GPS coordinates a world standard.

Still another consideration for using an absolute device like GPS is that map matching may be needed to smooth the motion of the display. The noise in the coordinates from the GPS receiver, if used directly, would cause the vehicle position to bounce with respect to the map. Map matching would remedy the problem.

By using more accurate coordinates one can plot more pleasing map displays, but pleasing map displays do not in themselves require accuracy. Most paper road maps pay no heed to accuracy and even purposely distort position to enhance legibility. The requirement to plot good looking maps taken with accuracy requirements for map matching implies rather high accuracy for Etak's applications.

We have discussed the need for relative accuracy in some cases, global accuracy in others, the use of different coordinate systems, geodetic, geocentric, as well as using different reference ellipsoids, and there are still other considerations that can be important such as projections. So coordinate accuracy is a complicated matter and its interpretation depends on the context. It is possible, perhaps common, to determine one's position very precisely and be completely lost, by misunderstanding coordinate systems (Ashkenazi 1986).

Generalization and Abstraction

To provide a regional context and help a driver navigate to distant destinations, a navigation system should provide various scale displays showing more or less detail for larger or smaller scales respectively. So a driver can zoom out to a small scale map and see the major limited access highways in an entire region and zoom in to a large scale to see all the streets in the immediate vicinity of the vehicle. Intermediate scale displays drop streets of lesser priority as the scale gets smaller: first local streets and trails vanish, then collectors, then arterials, and finally lesser highways. The presentation is also generalized in that curving roads are straightened at smaller scales.

This same prioritization scheme is useful in path finding. One prefers major highways to cross a region, for example. The classification of roads for use by computers is, at least in Etak's case, richer than that found on commercial street maps. This is because the use is different. One usually takes some time to read a paper map and the cartographer can depend on the reader to use remarkably good gestalt interpretation to find, say, a reasonable path out of a labarinthine neighborhood. The Navigator must present a map that the driver assimilates in a glance. The roads at various priorities must typically form a network to be helpful in finding routes to a destination, but for paper maps the need is much diminished.

Names, Addresses, Traffic Restrictions, etc.

Annotation on paper maps is usually called attribute data in digital maps. By either name the data is useful both for identifying map elements or instructions and for indexing maps and localities. The considerations here for navigation assistance are which classes of data to capture, degree of completeness, recency and accuracy. An important consideration is of course cost of data capture. In that regard field work is far more expensive than capturing data from available map sources. Reversing the viewpoint, the types of navigation assistance possible vary with whether or not field data capture is undertaken.

The classes of data we now capture at Etak are limited to those that do not require field work. Street names, addresses, relative importance of roads, major landmarks, and geographic area codes (city, ZIP, etc.) are usually available from public map sources and do not require field work. Turn restrictions and one-way flow restrictions are much less reliably available without field work and actual traffic sign content nearly always requires field work.

Giving reliable instructions often depends on having field collected data such as turn restrictions and this imposes a significant cost on the digital mapping effort. For our initial products we have avoided field work and made design decisions that take this restriction into account. Presenting the driver with a graphic display that gives enough information to safely and efficiently achieve a destination does not require knowing turn restrictions or one-ways much less sign content.

RETRIEVAL

Speed of retrieval, capacity of storage, and how to provide access to map data for application developers who may not wish to also provide navigation software or wish to provide alternative navigation software are the topics of this section. We have found this last subject to be very important, at least for Etak. We have licensed Etak navigation technology in the U.S., Europe, and Japan, and we must provide access to the map data so that different and independent developments can proceed. The requirement is the same even for Etak's own development of Navigators, Automatic Vehicle Location (AVL) systems, and map production facilities. The Map Engine, which is a software shell surrounding the data, is Etak's approach to providing access to the data.

Speed

Just keeping up with a vehicle's progress across the Earth is a non-trivial task for map retrieval software. Within the constraints of cassette tape, an 8088 CPU busily doing navigation and display calculations, and 256K bytes of RAM, the task is quite difficult. Some of the solution in the Navigator is in hardware: the tape has a capacity of 3.5 M bytes and operates at 80 inches per second. Some is in software and here topology is again important. Knowing what is neighboring the currently retrieved map elements is crucial to organizing the anticipatory sequence of reads from the tape.

The same software solutions will work well in CD/ROM storage. The capacity of CD/ROM is enormous by comparison: 500 M bytes per disk over 100 times greater than Etak's cassette tape. The speed is not so spectacularly better: average access time of 1 second, five to ten times faster than the cassette. So software that retrieves data from CD/ROM for onboard navigation may not need to be quite so clever but must still be faster than commercially available software in Geographic Information Systems (GIS).

Of course the same software works even better in the hard disk environment with 30ms average access times. In fact we have used the same software at Etak for our Automatic Vehicle Location (AVL) base stations, which are IBM/AT or similar computers with a graphics screen and mouse. So the dispatcher has the same functional capabilities as the driver but with much faster hardware. The base station has benefited greatly from the solution to fast map retrieval forced by the much more constrained vehicle environment.

Capacity

Digital maps require a great deal of memory. One EtakMap™, the cassettes containing a digital map and navigation software for the Navigator, covers the area of about two typical large scale paper road maps. Six cassettes cover, in an overlapping fashion the San Francisco Bay area. It takes ten in the Los Angeles/San Diego metro areas, three in Detroit, and five in New York, to give a few examples. EtakMaps contain both the map data for navigation and display and indexes for destination finding.

It appears to be possible to fit the entire US onto a single CD/ROM at the level of detail required for the Etak Navigator. It wouldn't make much sense to do so; the space would be better utilized for, say, Yellow Pages listing in categories like accommodations, restaurants, and services. However, it does indicate that when CD/ROM becomes available in a form that can survive in the harsh environment of a vehicle and at a price that is low enough, capacity won't be a serious constraint.

Map Engine

Not only is the data itself important but so is providing it to the various applications in an efficient and consistent manor. The applications include navigation, display, destination finding, and user specific applications. The database supporting high speed topological access to maps at various levels of abstraction using coordinates and annotation as the key for searching is, not surprisingly, complicated. To provide access to the data for other applications, other users, as well as to navigation and AVL, we have developed the Map Engine. It is a map-specific database management package accessed by subroutine calls. Examples, including some under development, of its capabilities are:

- Retrieve all map elements in a N-S E-W window
- Retrieve an open neighborhood around the current sub-map
- Retrieve a closed neighborhood ...
- Retrieve by feature name
- Set level of abstraction (generalization) for retrievals
- Retrieve a small subset of the map suitable for finding an optimal route to visit several destinations (which might return mixed levels of abstraction for distant points)
- Add, delete or change map elements
- Find a street address
- Find a intersection
- Find the nearest point on a street

By providing a software shell surrounding the data for a developer of applications has the same advantages that database systems provide. Certain parts of the job are done and debugged and proprietary advances are available prior to a patent issuing, which can be years sooner.

PRESENTATION

The extremes of styles of presentation are interactive graphic and driving instructions, written or voice. Intermediate stages are easy to imagine.

Graphics

A picture is worth a thousand words, but only if as much thought and skill went into creating the picture as the words. We endeavor to always present an uncluttered display with just a few labels, so that it is readable at a glance, but we choose what to label, the angle of the labels, the orientation of the map and the few other pieces of information to be as helpful as possible to the navigating driver.

In labelling streets we favor upcoming cross streets, higher priority streets and the streets at the destination. The map is oriented heading up so that what is ahead out the windshield is above on the screen. An arrow pointing to the destination with distance to go and a north arrow are the only other pieces of information on the screen.

The scale of the map determines the level of generalization. To always present maps of similar complexity at the various levels of generalization involves a significant effort in digital map production. In effect, digital cartographers are making driving choices for the driver in advance of actual driving; this is another place where we introduce a great deal of thought to make the picture worth the thousand words.

Not only is the resulting picture better than words, it is better than a printed map, by far. A new map is composed every few seconds designed precisely for the driver's current situation and displayed on the screen. It is a simple image, conveniently oriented (just how convenient is impossible to explain -- one has to use a Navigator to appreciate), with vast amounts of irrelevant data omitted. The information includes a star marking the destination (like the star of Bethlehem) which gets nearer to the car symbol as the vehicle approaches the destination; this provides a very warm feeling that you are making progress. Also included are nearby streets, which can give one the courage to exit a backed-up freeway and travel along surface streets. These are ways that digital maps are far superior to printed maps even for non-analytical uses.

Driving Instructions

Not everyone likes maps. There are even people who are intimidated by them. We have not seen this hostility transferred to the Navigator and this may be because the Navigator display does not suffer the intimidating clutter of many printed maps nor demand the geometrical intuition to reorient the map in your mind nor the difficulty of finding your place on the map.

Whatever the case, an alternative to a graphic display is a list of driving instructions. These can be given by voice, written on paper, or written on a screen as they are appropriate.

To generate reliable instructions, the digital map must contain turn restrictions to avoid impossible-to-follow directions but need not have a high level of coordinate accuracy, as mentioned above. To generate or modify the instructions on the fly both high speed retrieval and current location are needed. Missing a turn and encountering barricades or traffic jams are cases where modified instructions would be needed.

Both

There are many possibilities between the extremes and their demands on the digital map varies with the mix. One can present a graphic display with a preferred route highlighted. If a barricade prevents access to part of the route, the display has enough information for recovery. Audible tones or voice instructions as waypoints approach could be added to a display. The map display could be schematized even to the point of just indicating the direction of an upcoming turn with a diagram of the intersection, which would be especially useful in a many-street intersection. Such a simple display would be less demanding of coordinate accuracy (although if current location is maintained using map matching accurate coordinates would still be needed).

SUMMARY

Digital maps for navigation assistance must be topological for any significant level of sophistication. Navigation using map matching or displaying current position requires relatively accurate coordinates and satellite-based systems need globally accurate coordinates. Annotation helps to identify surroundings and to locate destinations. Fast retrieval of map data from copious storage is a requirement of tracking current location and generating dynamic driving instructions. Digital maps offer wonderful advances over printed maps for presentation of navigation data. First, far more useful and assimilable displays can be generated and second, instructions can be generated and presented graphically, in written form, or by voice.

REFERENCES

Ashkenazi, V. 1986, Coordinate systems: how to get your position very precise and completely wrong: Journal of Navigation, Vol 39, no 2.

Corbett, J. P. 1979, Topological principles in cartography, U.S. Bureau of the Census Technical Paper 48.

Neukirchner, E.P. and Zechnall, W. 1986, Digital map databases for autonomous vehicle navigation systems, Position Location And Navigation Symposium (PLANS), IEEE Las Vegas, NV.

Zavoli, W.B. and Honey, S. 1986, Map matching augmented dead reckoning, Proceedings of the 36th IEEE Vehicular Technology Conference, Dallas, TX.

ON GIVING AND RECEIVING DIRECTIONS: CARTOGRAPHIC AND COGNITIVE ISSUES

David M. Mark
State University of New York at Buffalo
Department of Geography
Buffalo, New York 14260

ABSTRACT

Analysis of 20 sets of navigation directions, prepared to accompany invitations to events, is used to examine both cartographic and cognitive issues related to intra-urban navigation. First, maps are more commonly used (18 of 20 examples) than are verbal-procedural instructions (7 of 19 cases), even when a common trip-origin can be assumed. It appears that correct street orientation (and a north-arrow) are highly desireable features of a navigation aid, whereas neither an absolute scale nor even correct relative scale is important. Landmarks were present on 14 of the 18 maps; these included traffic lights (7 maps), fast-food outlets and gas stations (5 each), supermarkets (4), and schools, shopping centers, and convenience stores (3 each). Implications of these results, both for automated in-car navigation aids and for acquisition of spatial knowledge, are presented.

INTRODUCTION

When people give directions, they may draw maps, give verbal directions (printed or spoken), or use parts or copies of published maps (with or without annotation). The nature of direction-giving will differ, depending on whether the communication is in person or remote, one-way or two-way, written or aural, etc. This paper presents a general cognitive model for spatial learning, and the implications of that model for direction-giving and direction-following. Then, examples of directions (given on paper) intended to be used for navigation to novel destinations are analyzed. Finally, the work is placed in the context of designing computerized navigation aids for drivers, a topic of considerable interest to cartographers and the subject of sessions at recent Auto Carto meetings (Cooke, 1985; Streeter, 1985; White, 1985; Mark and McGranaghan, 1986).

CONCEPTUAL FRAMEWORK

The conceptual basis for this and related studies of direction-giving and way-finding lies in the new inter-discipline of **cognitive science.** Cognitive science is "a new and dynamic field that emerged from the unlikely marriage between cognitive psychology and computer science" (Couclelis, 1986, p. 2). In cognitive science, "cognitive functions such as problem-solving, pattern recognition, decision-making, learning, and natural language understanding are investigated by means of computer programs that purport to replicate the corresponding mental processes" (Couclelis, 1986, p. 2). Spatial learning has been a recurring theme in cognitive science (for example, Kuipers, 1978; Riesbeck, 1980; Thorndyke and Hayes-Roth, 1982). Recently, Mark and McGranaghan (1986) reviewed relevant literature in cognitive science, and proposed that this forms a useful approach to the problem of providing navigation assistance to drivers.

A Model for Spatial Knowledge Acquisition

Benjamin Kuipers has developed a powerful computational model for spatial knowledge. In particular, Kuipers' model is concerned with the processes by which one learns about large-scale (geographic) space. The model was introduced in Kuipers' 1978 paper in **Cognitive Science**, and was refined and expanded in a series of other papers. Mark and McGranaghan (1986) have proposed that Kuipers' model of spatial knowledge **acquisition** also forms an appropriate theoretical basis foɪ studies of the **communication** of spatial information for navigation and other purposes.

Kuipers' model organizes spatial knowledge into three major categories. First, sensorimotor procedures consist of sets of actions, and their sequence, that are required to travel from one place to another (Kuipers, 1983b, p. 1). Second, topological relations represent "knowledge of non-metrical properties of the external environment, such as containment, connectivity, and order" (sequence) (Kuipers, 1983b, p. 1). Finally, metrical relations encode "knowledge of, and the ability to manipulate, magnitudes such as distance, direction, and relative position." (Kuipers, 1983b, p. 1). For convenience, Mark and McGranaghan (1986) referred to these as **procedural**, **topological**, and **metrical** knowledge, respectively.

Procedural knowledge is based on two types of objects, **views** and **actions**. A **view** is the set of sensory inputs available at a particular place and orientation; it is important that one can determine whether or not two views are the same. An **action** is a motor operation (such as a move or a turn) that changes the current view. As one travels through large-scale space, one "sees" a series of views; some of these views are associated with actions such as turns from one street to the other. **Routes** can be remembered as collections of "view-action" pairs. If a person remembers "(view-action)-view" triples, routes can be recreated in the mind, and described to others.

Kuipers' model proposes that many people are able to generalize from procedural knowledge of routes, and build **topological knowledge** of large-scale space. At this level, a **place** is identified as the cycle of views after repeated turns at a point; places may be common to more than one route. People with spatial knowledge at a topological level will usually know on which side of a major street (path) some known place lies, and will be able to plan new routes between places. However, the orientations of paths may be distorted, and places are not fitted into an overall coordinate system. Sketch maps produced by people with this level of knowledge may be distorted, but often will be useful and fully functional for spatial navigation. Also, regions at one level of abstraction may be equivalent to places at another level.

Kuipers' model proposes that some individuals integrate spatial knowledge acquired through navigation and produce spatial knowledge at the metrical level; at such a level, spatial knowledge is placed into the framework of a cartesian coordinate system. Mark and McGranaghan (1986) observed that access to graphic metrically-correct maps almost certainly plays a key role in the learning of spatial information at this level. Research by Thorndyke and Hayes-Roth (1982) appears to support this contention, although their experiments were conducted inside large buildings, rather than in an urban street network.

Kuipers' Model and Spatial Navigation

Mark and McGranaghan (1986) claimed that it is useful to classify various forms of spatial information for navigation according to the three categories of spatial knowledge proposed by Kuipers. Verbal directions for getting from one place to another, presented in words either spoken or printed, represent information at a procedural level. In a vehicle, procedural knowledge could also be provided to the driver in non-verbal form, by means of signals produced either by a human navigator (for example, pointing when it is appropriate to turn), by an on-board computer, or by electronic signposts. Sketch maps with distortions (deliberate or inadvertent) represent a topological level of spatial information. Road maps and other planimetrically-correct maps represent a metrical level of information.

Clearly, procedural (sensorimotor) knowledge **must** be available in the mind of the traveller so that order the traveller can make the decisions necessary to get from one place to another. If navigation information is provided at **any** level other than procedural (for example, in the form of a graphic map on paper or on a computer-graphics display device), then the traveller or navigator must do work to determine the relevant procedural instructions. This takes time and effort, may distract from other driving tasks, and may be subject to error.

Graphic Maps or Verbal Directions?

Graphic maps have been so much the dominant form for the representation of spatial information in support of navigation that the relative effectiveness of the map in this context has hardly been questioned. However, in addition to the cognitive theory presented above, there is empirical evidence that procedural directions may be useful and effective.

Astley (1969) reported the results of questionnaire surveys of 300 British road-users. Most of those surveyed used maps. However, "nearly half of the respondents used written route guides" or some kind; "the majority wrote their own from maps" (Astley, 1969, p. 130). One can infer that these travellers felt that written-procedural directions are easier and/or better to use during trips than are maps.

In perhaps the only experimental investigation of the relative effectiveness of navigation aids for drivers, Streeter and others (1985) found that drivers appear to navigate more effectively when given verbal (vocal) directions, rather than a graphic map. Streeter and others compared four methods for receiving navigation aid during automobile driving. These methods are: (1) standard road maps; (2) customized road maps (north at top); (3) verbal instructions from tape recorder; and (4) a combination of methods (2) and (3). Performance of test subjects was evaluated in terms of travel time, number of errors, and other measures. They found, not surprisingly, that method (1) (the "control" condition) produced the poorest performance. However, the best performance was observed when the subjects had only the tape recorder to guide them. (This method involved a customized tape recorder with two buttons: one to repeat the last instruction, and the other to play the next one.) Significantly, the "customized map" group (method 2) had the second-worst performance level. It also seems that providing the subjects with maps in addition to the tape recorders (method 4) detracted from performance given the tape recorder only. Perhaps the map constituted a distraction, reducing the ability of the subject to concentrate on the tape; alternatively, by providing a means to recover from errors, it may have reduced the perceived **need** to concentrate on the tape.

The main result of Streeter's experiment appears to confirm the implication that Mark and McGranaghan (1986) drew from Kuipers' model, namely that provision of navigation information at the procedural level should be easier to assimilate than would be graphic topological or metrical information. Mark (1985) presented a method for computer generation of routes which were simple to describe, at the possible expense of greater route length.

DATA

The directions (navigation-aid information) analyzed in this study were associated with invitations to parties and other events. These were "natural" directions actually prepared and distributed by event organizers. Preparers had time to think about the directions, to try again if the directions were not adequate, etc.; thus they might be expected to be different from directions produced more spontaneously. Only a few of the directions were prepared by cartographers. Some present only verbal directions, others include only maps. Still others are annotated maps (integrated graphic/verbal instructions), or include both maps and verbal directions, either of which could stand alone. Maps range from photocopies of printed road maps, to geometrically-correct maps showing only relevant information, to highly schematic diagrams which really illustrate only the topology of the route to be followed.

All 20 sets of directions analyzed were "mass-produced" and distributed along with the invitation and/or announcement of the event. Thus, the individuals preparing the directions could assume details regarding neither the recipient (knowledge of the area, navigation skills, etc.) nor the location of the trip origin. However, all directions were prepared in relation to a **single** event, and thus were intended for a particular **group** of people (the invitees). Thus more general characteristics and origins might have been known to the preparers of the directions.

The writer was a the recipient of 17 of the 20 sets of directions. As one might expect, many of the directions were prepared by cartographers (5 cases) and other geographers (9 cases); this introduces a potential for bias, since we might expect navigation aids produced by such people to be somewhat more "professional" than would be typical in our society. (However, readers who know cartographers and/or geographers will probably confirm the writer's impression that geographers and cartographers are outstanding in neither their navigation abilities nor their direction-giving abilities!) Results must be interpreted in light of this distribution.

RESULTS

All of the 20 sets of directions examined in this study were reproduced by photocopy, and consisted of black marks on paper; in only one case was any color added (the suggested route was drawn in red by hand on each map). Of the 20 examples, 18 included maps; verbal-procedural directions were included in 7 cases, accompanying maps in 5 of them. Thus 13 examples had maps as the only source of navigation information. The inclusion of verbal-procedural directions seems to be more frequent among non-geographers (see Table 1), but two of the "map + verbal" examples were prepared by the **same** non-geographer; without her second set of directions (prepared two years after the first set, in relation to a different destination), the association would

TABLE 1

Association Between Form of Directions
and Background of Preparer

Preparer:	Cartographer	Geographer	Other
Map only	4	6	3
Map + verbal	1	1	3*
Verbal only	0	1	0

* includes two examples by same person

be weak, if present at all. Since maps were present in 18 cases, this section will concentrate on an analysis of the characteristics of these maps, under several sub-headings.

Orientation and scale. Text present on the paper clearly indicated an "up" orientation for each map. On all but two of the maps, north was in this "up" direction; in the other two, the orientation was "heading-up", that is, the goal was at the top of the map, and the typical trip origin was toward the bottom. Furthermore, north-arrows were drafted on 13 of the 18 maps (72 %). Two of the 5 maps without north-arrows were prepared by non-geographers, the other 3 by geographers who do not specialize in cartography. (Interestingly, neither of the two "south-up" maps had a north-arrow or any other explicit indication of cardinal directions!) In contrast, only 3 maps (17 %) included any explicit indication of scale; in two cases, a scale bar was drawn on the map, and in the other, two important road segments were labelled with their lengths in miles. Of the remaining 15 maps, only one other map explicitly stated: "scale variable"; users without local knowledge would have no way of knowing whether the other 14 maps were "to scale" or not, and what that scale might be.

Landmarks. Considerable evidence suggests that landmarks are very useful in navigation. Data from the current study supports this, since 14 of the 18 maps included either traffic lights, other landmarks, or both. In fact, general landmarks were included on 11 of the maps; between 1 and 7 landmarks other than traffic lights were present, with a mean of 3.1 instances per map using them. The landmarks are classified and tabulated in Table 2.

TABLE 2

Landmarks Used on Maps Studied

fast food outlet	5
service station	5
supermarket	4
school, campus	3
shopping center	3
convenience store	3

11 other landmark types appeared once each:

church, airport, funeral home, restaurant, doughnut shop, car dealer, bar, railroad, department store, hospital, hotel.

Traffic lights were included on 7 of the maps; four different symbols were used (see Figure 1), although only one of the maps included an explanation of the symbol used. Note that traffic lights are frequently used as landmarks in spontaneous, two-way direction giving.

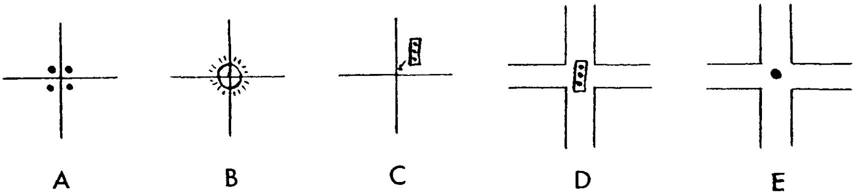


Figure 1: Traffic-light symbol A was used twice, and the others once each (C and D are the same symbol, used with single-line streets and "wide" streets).

Geometry. The geometry of each map was assessed qualitatively by the writer. Five of the maps were geometrically correct (having been photocopied or traced directly from roadmaps), and another 3 were hand drawn maps corresponding closely with the correct geometry. Nine of the remaining 10 maps contained substantial distortions of geometry (presumably inadvertant), but were topologically adequate. The tenth map used the technique of jagged breaks in wide roads to indicate scale variation produce by the omission of sections of roadway; this map attached lengths (in miles) of two road links.

In 7 of the 9 distorted maps and in the "interupted map", orientations of streets generally were correct, but distances showed large distortions. For 7 of the distorted maps, the distortions probably were not great enough to confuse a typical navigator. However, the geometry of the remaining 2 maps was highly distorted, and could easily have confused navigators who were not sufficiently familiar with the area to ignore these problems. Consider the example shown in Figure 2: Bailey Avenue actually is straight, and it appears straight on the ground; however the sketch map would lead one to expect major curves, and thus could produce cognitive dissonance in the mind of a navigator. In the other distorted map, angles were more-or-less correct, but relative distances were **greatly** distorted. Interestingly, correct geometry and inclusion of verbal-procedural directions showed a **positive** association (see Table 3). That is, of the 5 maps which

TABLE 3

Association Between "Correct" Geometry and Verbal-Procedural Directions

Verbal Directions?	Correct Geometry? Yes	No
Yes	4	1
No	4	9

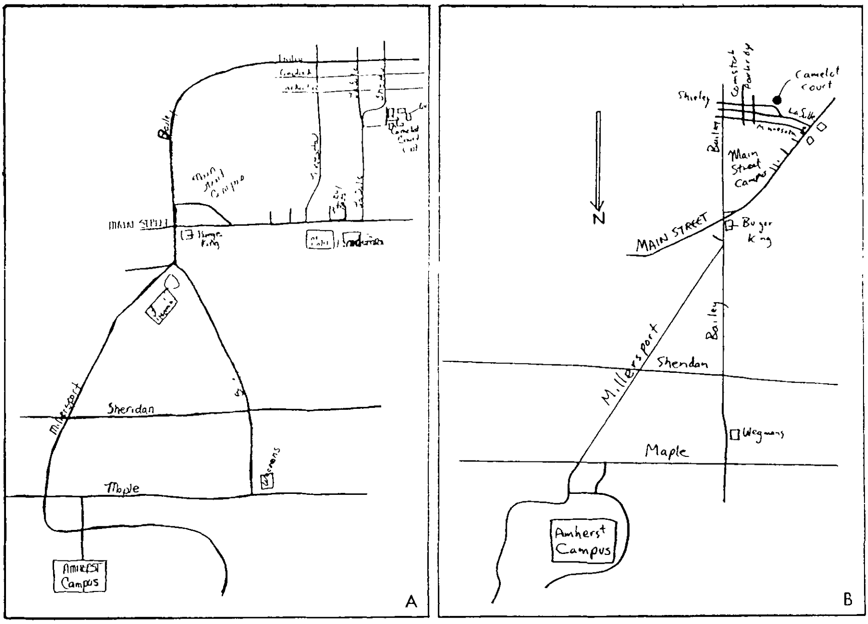


Figure 2: An example of a distorted sketch-map (A), and the same streets traced from a road map (B). South is at the top.

also included verbal-procedural directions, 4 had a "correct" geometry; of the 9 maps with substantially distorted geometries, only one included verbal directions.

Route Advice. As noted above, 5 of the 18 maps included verbal-procedural directions. In 3 examples, the verbal directions and the maps were equally prominent on the paper; for the other 2, the verbal directions were present as marginal information. In 3 sets of verbal directions presented with maps, as well as in the single example of verbal-only directions, a common origin for all trips was assumed; this was located at a recognizable point between most trip origins and the destination. In each case, it is the writer's opinion that almost all people invited would approach the destination from the assumed direction. In one of the two remaining cases, two different first steps were included to get travellers to a common point; in the other, two separate sets of directions were presented, one to be used by those approaching from the east, the other from the west.

Five of the 18 maps indicated a suggested route through the use of cartographic symbols; four included arrows indicating the suggested route, and the fifth drew the route in color on an otherwise monochrome map. Three of these maps were also accompanied by verbal directions, suggesting the same route using two different methods. Three of the maps with arrows placed them on all street links on the suggested route(s), whereas one placed arrows only at turns.

DISCUSSION

It would be a mistake to assume that methods used in informal cartography, primarily by untrained map-makers, provides a model of the ideal navigation-aid system. (This was pointed out by Judy Olson in comments during a presentation by the writer in East Lansing, Michigan, October 1986.) Clearly, a trained cartographer should be able to produce graphical navigation aids which are better and more effective than most of the ones analyzed here. Nevertheless, the results give some indications of characteristics which motorists find desireable in maps; at the very least, illustrate the sorts of "errors" or distortions which can be **tolerated** in a road map.

First, it is very clear that most people feel that **orientation** is essential in a graphic navigation aid. Even though street names and street patterns would provide orientation cues to navigators familiar with the area, and although all but two of the maps followed the cartographic tradition of a "north-up" orientation, fully 72 percent of the map-makers (13 of 18) added a north-arrow to the map. Conversely, **scale** seems unimportant, since only 3 of the 18 maps had any scale indication.

A purely topological spatial model (Kuipers, 1983b, p. 1) would contain neither orientation nor scale, whereas both would be present in a geometrical model of spatial knowledge. The present results suggest that an intermediate level of spatial knowledge is needed for intra-urban navigation: this intermediate level includes at least a generalized indication of orientation, but no precise indication of scale. The presence of an intermediate level is further supported by the fact that only 2 of the 16 maps analyzed included significant distortions in street orientations, whereas 7 included substantial errors of relative distances. This result also suggests that orientation information may be acquired before distance information during spatial learning; this would be an interesting area for further research.

Finally, the navigation aids analyzed in this study confirm the centrality of maps in this context. In a static medium (print on paper), verbal directions are inadequate for most situations for at least two reasons: (1) both the origin and the destination of the particular trip must be know (the map, on the other hand, allows the navigator to generate directions from **any** trip origin); and (2) printed verbal directions do not allow for straight-forward error recovery (whereas on a map, a new route can always be planned from the current location to the goal). It is important to note, however, that these weaknesses of verbal-procedural directions would not apply if the directions were produced in real time by an on-board computer with a street-network data-base and a location facility. The relative importance of procedural and graphic navigation output from computerized vehicle navigation aids appears to be an open question.

REFERENCES

Astley, R. W., 1969, A note on the requirements of road map users. The Cartographic Journal 6, p. 130.

Cooke, D. F., 1985, Vehicle navigation appliances: Proceedings, Auto-Carto 7, 108-115.

Couclelis, H., 1986, Artificial intelligence in geography: Conjectures on the shape of things to come: Professional Geographer, 38, 1-11.

Kuipers, B., 1978, Modeling spatial knowledge: Cognitive Science 2:129-153.

Kuipers, B., 1983a, The cognitive map: Could it have been any other way? in Pick, H. L., Jr., and Acredolo, L. P., editors, Spatial Orientation: Theory, Research, and Application. New York, NY: Plenum Press, pp. 345-359.

Kuipers, B., 1983b, Modeling human knowledge of routes: Partial knowledge and individual variation: Proceedings, AAAI 1983 Conference, The National Conference on Artificial Intelligence, pp. 1-4.

Mark, D.M., 1985, Finding simple routes: 'Ease of description' as an objective function in automated route selection, Proceedings, Second Conference on Artificial Intelligence Applications (IEEE), Miami Beach, December 11-13, 577-581.

Mark, D. M., and McGranaghan, M., 1986, Effective provision of navigation assistance to drivers: A cognitive science approach: Proc., Auto Carto London, 2, 399-408.

Riesbeck, C. K., 1980, 'You can't miss it': Judging the clarity of directions: Cognitive Science, 4, 285-308.

Streeter, L. A., 1985, Comparing navigation aids for computer-assisted navigation: Paper Presented at Auto Carto 7 (not published in proceedings).

Streeter, L. A., Vitello, D., and Wonsiewicz, S. A. 1985, How to tell people where to go: Comparing navigational aids: International J. Man/Machine Studies, 22, 549-462.

Thorndyke, P. W., and Hayes-Roth, B., 1982, Differences in spatial knowledge acquired from maps and navigation: Cognitive Psychology, 14, 560-589.

White, M., 1985, Building a digital map of the nation for automated vehicle navigation: Proceedings, Auto-Carto 7, p. 570 (abstract only).

RESEARCH INTO ELECTRONIC MAPS
AND
AUTOMATIC VEHICLE LOCATION

Edward J. Krakiwsky, Hassan A. Karimi,
Clyde Harris and Jim George
Department of Surveying Engineering
The University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4

ABSTRACT

This paper begins with an overview of possible land-based applications of electronic maps and Automatic Vehicle Location (AVL) systems in the commercial, civil and military sectors. The desireable software and hardware characteristics of these AVL systems are defined. Existing AVL systems are overviewed. A prototype AVL system, named AVL 2000, is being developed and tested at The University of Calgary in order to help define problems and identify solutions, primarily with on-road, land applications. The hardware segment consists of a Trimble 4000S GPS satellite receiver and a MACPLUS micro-computer with a graphics display all mounted in a van. The software segment consists of a control program, map and route data bases, as well as data bases for auxiliary and collected data. A "best route" determination scheme is part of the AVL 2000 prototype system. A learning capability employing artificial intelligence techniques is the important feature of future models of the AVL 2000 system. Experience with the AVL 2000 prototype system is discussed.

INTRODUCTION

The mid 80's marked the "dawn" of Automatic Vehicle Location (AVL) systems; the 1990's will be the decade in which AVL systems will "blossom"; while the year 2000 will mark the beginning of the age of widespread acceptance and usage of AVL systems.

An AVL system allows a land based user to:
(a) position a vehicle using signals from satellites and information from on-board differential positioning devices;
(b) plot the position on a CRT or flat panel display;
(c) call up a digitized-electronic map of the area and see the vehicle's position relative to a desired location(s); and
(d) obtain instructions (visual and audio) using an expert system on how to proceed from the present location to the desired location (Figure 1).

Getting from location A to location B, and then to location C, and so on, in an optimal manner will soon be possible - getting lost may be a thing of the past.

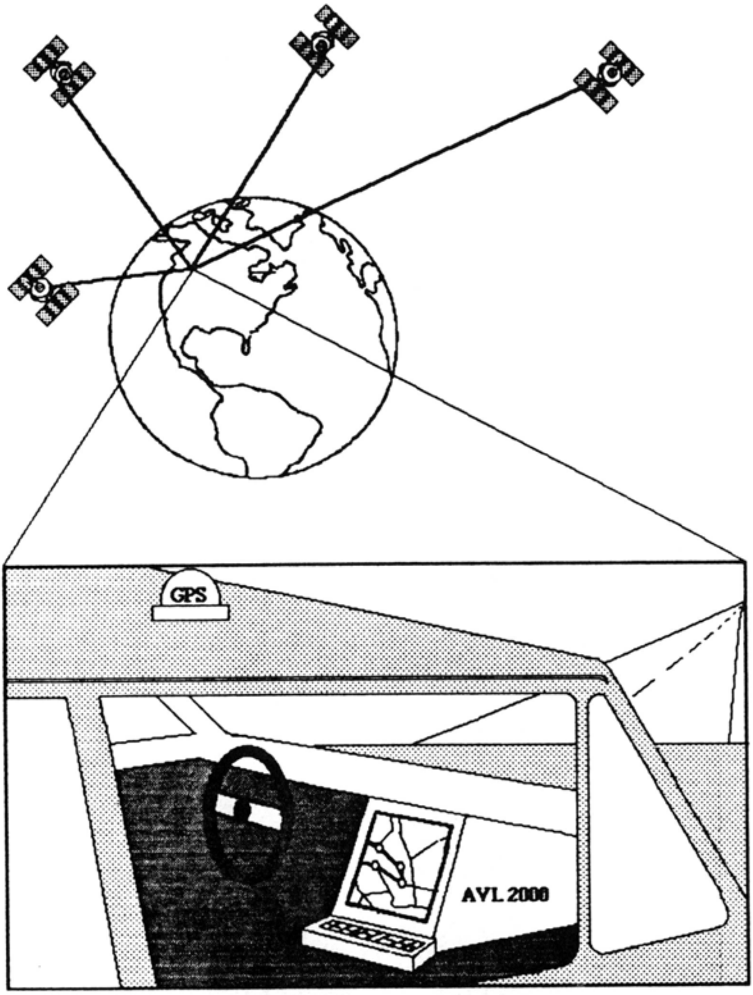


Figure 1. Concept of an Automatic Vehicle Location System

Drivers of police and fire prevention vehicles, ambulances, truck and taxi fleets, courier delivery fleets, farm vehicles, vehicles collecting data (e.g. geophysical), civic vehicle fleets, and even private automobiles will find AVL systems indispensable. The end result will be an enormous saving of time and energy - leaving vast sums of money to help solve other problems in our society.

APPLICATION OF AVL SYSTEMS

AVL systems can find applications in the air, marine and land environments. In this paper we restrict ourselves to use on land. Summarized in Table 1 are applications in four major sectors, namely commercial, civic, private and military. To appreciate how users in these sectors can benefit from an AVL system is best achieved by simply letting one's imagination take over.

Table 1

APPLICATION OF AVL SYSTEMS (LAND-BASED)

COMMERCIAL

- delivery and collection fleets
- taxi fleets
- position-based land information gathering vehicles (on and off road)
- automobile associations
- farm trucks and implements

CIVIC

- police and fire prevention vehicles
- waste and refuge removal fleets
- ambulances

PRIVATE

- passenger car
- off road recreational vehicles

MILITARY

- on and off road vehicles

The scientific approach to studying the application of AVL systems is to regard the activities of Table 1 as problems in optimization. Leuenberger [1969] suggests that there are at least four distinct optimization problems: (1) estimation; (2) allocation; (3) planning and logistics; and (4) control. In each of the above, some objective function is either minimized or maximized. Let us discuss each of the above within the context of AVL systems and applications.

Estimation concerns itself with the solution of a set of unknown parameters (e.g. AVL coordinates) under the condition that the quadratic form of the errors (e.g. on the observations to satellites) is a minimum. Allocation deals with the distribution of a set of resources (e.g. trucks in a fleet) such that a given amount of work can be done at minimum cost. Planning and logistics involve the placement and movement of resources (vehicles) such that minimum time or cost is incurred. Control is the guidance of an entity (vehicle) along a given path (route). Clearly, optimization is a major part of an AVL system.

AVL systems employ one or more of the above. For example, some application areas, such as those dealing with fleets, are dispatch oriented and, clearly, would involve all four optimization problems. At the other end of the spectrum, a private passenger vehicle would involve only estimation (position) and control (route direction). Nevertheless, the types of problems that need to be solved in AVL systems are clear. This in turn dictates the array of components that must be part of AVL systems.

AVL COMPONENTS AND THEIR FUNCTIONS

AVL systems are clearly an assembly of technologies [Skomal 1981]. These include: (1) positioning systems; (2) a computer-microprocessor; (3) input devices; (4) output devices; (5) map storage devices; and (6) management and computational software (Figure 2).

Positioning systems can include Loran-C, Transit satellite and GPS receivers for point (absolute) positioning; for relative positioning, differential wheel movement devices, and odometer-compass combinations. Point positioning devices can be used alone, or can be integrated with relative positioning devices. They have a symbiotic relationship - when point positions are not available (say due to satellite masking or outages), relative positioning devices are used to interpolate positions. In the case that relative positioning devices have operated alone for some time, their drifts can be corrected - updated by point positioning devices.

Computers and microprocessors are needed to process and send various kinds of data between the multitude of devices. These may be self-contained units in each vehicle, or reside at the central dispatch station. The latter are larger and have more capacity. Typically, self contained computers on board vehicles are small PCs containing limited memory. Microprocessors of comparable size are one order of magnitude lower in price and carry out well-defined, specific functions.

Input devices vary according to what unit we are speaking about. The vehicle unit can have a (i) keyboard-pad; (ii) finger touch control; (iii) transmitter-receiver; or even (iv) voice input. The base unit for dispatch systems can have the following input devices: (i) transmitter-receiver; (ii) radio; (iii) keyboard-pad. Roadside input units usually have an inductive loop buried in the road surface and a transmitter-receiver.

Output devices for the vehicle itself include the following: (i) vector display CRT; (ii) voice synthesizer; (iii) video display; and (iv) a transmitter-receiver. The base unit for dispatch systems has the same units as the vehicle itself plus a radio to transmit and receive information to the vehicles. Roadside output units are the same as input units.

Map storage and management can be self-contained in the vehicle itself and be in the form of an electronic map inside the computer, on a video disc, a photographic map, or special map sheets. Maps can also originate from a control database and access may be direct, thus yielding a true electronic map, or it may be public access through a centrally located terminal.

Software is needed to perform many specialized functions. Some of these include: (i) speech algorithm; (ii) database and display algorithm; (iii) zooming algorithm; (iv) optimized route algorithm; (v) auxiliary information capability; and even (vi) an expert system capability.

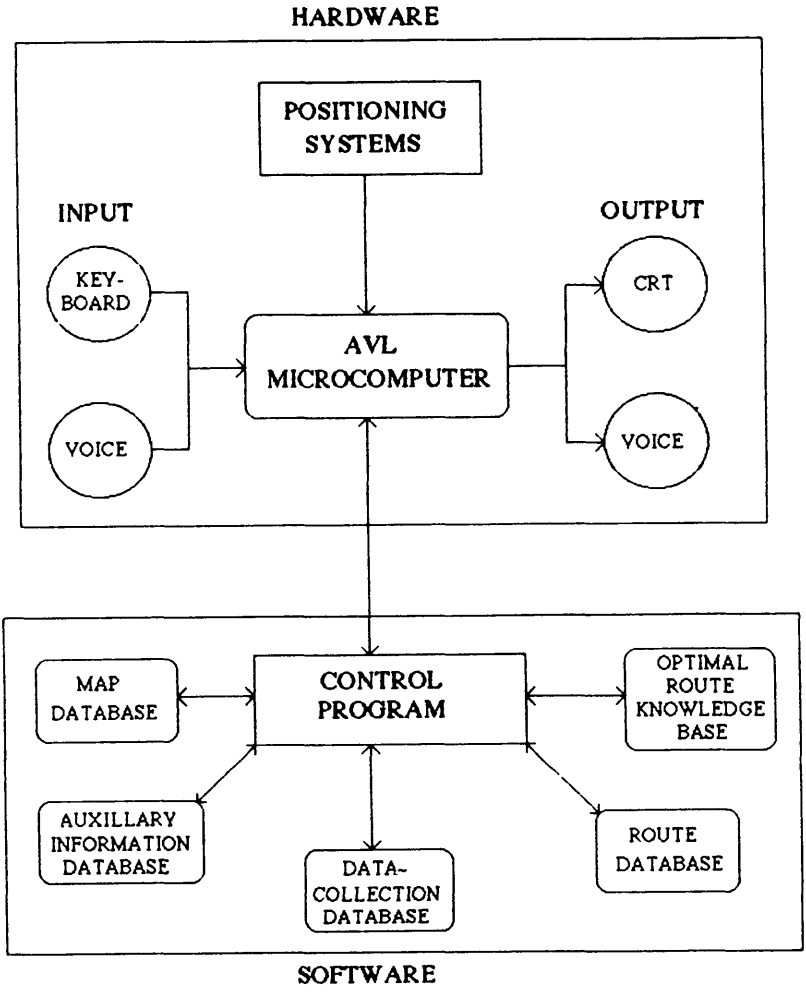


Figure 2. Components of an Automatic Vehicle Location System

EXISTING AND DEVELOPING AVL SYSTEMS

The authors have researched the literature and found that a total of at least 18 AVL systems exist and are under continuous development and improvement (Table 2). This is ample evidence for the statement made earlier that the 1990's will be a period in which AVL systems will blossom. They will be highly specialized and hence aimed at a certain spectrum of the market. No one system will be capable of serving the entire market.

The one component of AVL systems that is more or less a common denominator for on-road applications is the electronic map. Nevertheless, even this component will

EXISTING AND DEVELOPING AVL SYSTEMS	HARDWARE																					
	NAVI-GATION			COMP-UTER		INPUT FACILITIES									OUTPUT FACILITIES							
						VEHICLE UNIT				BASE UNIT			ROAD		VEHICLE UNIT			BASE UNIT			ROAD	
	OPEN LOOP D.R.	CLOSED LOOP D.R.	GPS SATELLITE SYSTEM	SELF-CONTAINED	LOCATED IN BASE STATION OR ROADSIDE UNIT	KEYBOARD/PAD	FINGER TOUCH CONTROL	TRANSMITTER/ RECEIVER	VOICE INPUT	TRANSMITTER/ RECEIVER	RADIO	KEYBOARD/PAD	INDUCTIVE LOOP	TRANSMITTER/ RECEIVER	VOICE SYNTHESIZER	VIDEO DISPLAY	TRANSMITTER/ RECEIVER	TRANSMITTER/ RECEIVER	VIDEO DISPLAY	RADIO	INDUCTIVE LOOP	TRANSMITTER/ RECEIVER
Etak Navigator		✓		✓		✓										✓						
CARGuide				✓		✓									✓	✓						
Flair	✓	✓			✓	✓		✓		✓	✓	✓						✓	✓	✓		
VDO Citypilot		✓		✓		✓										✓						
Navigator				✓		✓									✓	✓						
Chrysler : CLASS			✓	✓			✓									✓						
Buick Questor			✓	✓																		
Honda : Gyrocator	✓			✓												✓						
Nissan : CUE - X			✓	✓												✓						
Inductive loop based Guidance System					✓	✓		✓					✓	✓		✓					✓	✓
Experimental Systems					✓			✓						✓		✓						✓
Auto - Scout					✓	✓		✓						✓		✓						✓
Wootton Jeffrey				✓		✓									✓							✓
ROUTE - TEL					✓	✓																
Route Systems					✓	✓																
Magnavox	✓		✓		✓	✓	✓	✓		✓	✓					✓	✓	✓	✓	✓		
AVL 2000 (prototype)			✓	✓		✓										✓						
AVL 2000 (1st generation)	✓	✓	✓	✓		✓	✓		✓						✓	✓						

Table 2. Existing and Developing Automatic Vehicle Location Systems

EXISTING AND DEVELOPING AVL SYSTEMS	MAP STORAGE						SOFTWARE					
	SELF-CONTAINED DATABASE				CENTRAL DATABASE							
	ELECTRONIC MAP	VIDEO DISKS	PHOTOGRAPHIC MAPS	MAP SHEETS	ELECTRONIC MAP	PUBLIC ACCESS TO CENTRALLY LOCATED TERMINAL	SPEECH ALGORITHM	DATABASE / DISPLAY MANIPULATION ALGORITHM	ZOOMING ALGORITHM	OPTIMAL ROUTE ALGORITHM	AUXILIARY INFORMATION CAPABILITY	EXPERT SYSTEM ALGORITHM
Etak Navigator	✓							✓	✓			
CARGuide	✓						✓	✓	✓	✓		
Flair					✓			✓	✓		✓	
VDO Citypilot				✓								
Navigator	✓						✓	✓		✓	✓	
Chrysler : CLASS		✓						✓	✓			
Buick Questor	✓							✓				
Honda : Gyrocator			✓									
Nissan : CUE - X												
Inductive loop based Guidance System					✓			✓		✓	✓	
Experimental Systems								✓	✓	✓		
Auto - Scout								✓				
Wootton Jeffrey	✓							✓				
ROUTE - TEL						✓				✓	✓	
Route Systems						✓				✓	✓	
Magnavox					✓			✓		✓		
AVL 2000 (prototype)	✓							✓		✓		
AVL 2000 (1st generation)	✓	✓					✓	✓	✓	✓	✓	✓

Table 2. Existing and Developing Automatic Vehicle Location Systems (continued)

need extensive customizing. This customizing will depend upon the special feature(s) of the particular AVL system. To illustrate this point, witness the AVL 2000 system (Table 2). Two levels of customizing had to be done: the first, occurred in developing the map database itself from a transportation network database, and a second level occurred in establishing the route database so that minimum route calculation could be done in real time in the vehicle.

So called public-based AVL systems are illustrated by the Routes System (Table 2). Note, the characteristics and components of this system are quite different from, say, the Buick Questor system which is essentially self-contained.

THE AVL 2000 SYSTEM

The AVL 2000 system is an in-vehicle real-time system which utilizes an integrated positioning system to, first, locate a moving vehicle and then have its position superimposed on a digital route map displayed on a CRT. One of the main features of the AVL 2000 system is a real-time optimal route selector [e.g., Dijkstra 1959], that is, the determination of the "best" route between the starting point and the destination point. The criteria for "best" route depends on the type of application [Karimi et al. 1987].

The first phase in the development of the AVL 2000 system was to assemble a prototype from existing hardware. The main challenge faced in developing the prototype system was system integration, which included interfacing the hardware components and developing software. The prototype is a microcomputer-based AVL system and an algorithmic approach was used in its software development. Illustrated in Figure 3(a) is the AVL 2000 system prototype configuration.

The first generation system is to be a product model. The design includes the integration of a GPS point positioning receiver and dead-reckoning device. An integrated positioning system would provide continuous and reliable positioning. Figure 3(b) shows the first generation AVL 2000 system configuration.

The second generation of the AVL 2000 system is conceived as being an AVL system which will use the latest advancement in both hardware and software components. As such it will be a microprocessor-based AVL system, and its software will be based on heuristic principles instead of being entirely algorithmic. It will also be an intelligent-customized AVL system in which the state-of-the-art design and architecture will be used; namely it will be VLSI based. Its configuration is shown in Figure 3(c).

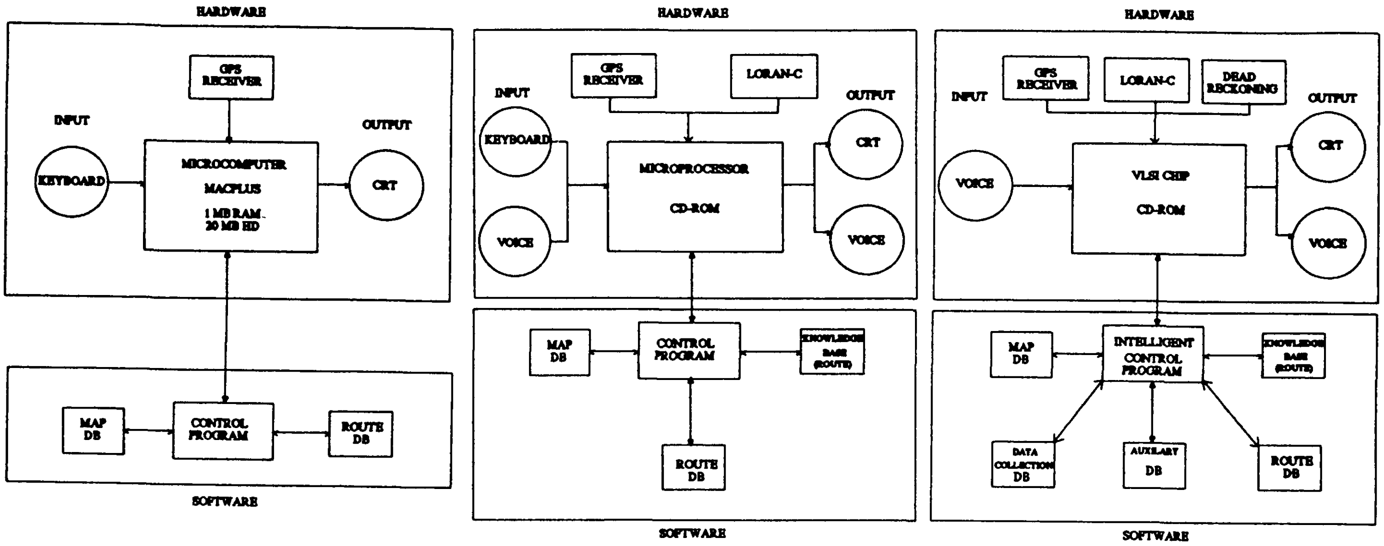


Figure 3. AVL 2000 Systems

PRACTICAL EXPERIENCE WITH AVL 2000 (PROTOTYPE)

The aim of this research was to build a prototype system as quickly as possible and get some practical experience at the earliest possible stage of the research. This we have done and discussed below are some of the problems encountered and solutions we have formulated. Experience has been gained with the hardware, software and, as important, with the digital map information.

Interfacing of the Trimble 4000S GPS receiver and MACPLUS was readily accomplished via a standard serial port (RS232). This configuration was then mounted in a van with power supplied from a 12 volt battery and from the vehicle using an alternator.

Digital-electronic maps for the Calgary test area are not available. Two sources of information were used to create a customized electronic map for AVL purposes. The first was a coordinate file of all road intersections in the Alberta Transport Link-Node Network [Yeung 1986]. Auxiliary information included: class of road; speed; length of link; etc. The second set of information was a coordinate file of 72 municipal polygons of the Province of Alberta. This latter information was needed in order to divide the province into manageable parts for the real-time operation of the minimum route algorithm.

To customize the data for the AVL prototype, the boundary and link files were merged. Each link was checked against each polygon to determine if it was within that polygon. This was done by first checking the maximum and minimum limits of the polygon, then using a point in polygon algorithm (number of times link crosses polygon boundary) to exactly determine if the link is within the polygon. The polygon ID was then attached to the link. If the link was within two polygons, an additional node was placed at the boundary to split the link so that no link is within two polygons. The large-merged data file was then split into one data file for each polygon to enable faster access.

Displayed in Figure 4 are the primary routes contained in one of the 72 municipal district polygons. Contained therein is the City of Calgary. Also shown in the figure is the best route selected between beginning and destination points.

Shown in Figure 5 is a hard copy of the CRT image of the AVL 2000 system. It is a zoom-view of a portion of the optimum route. Note a major turn is indicated and notice is given to the fact that the driver has left the "best" route.

Several field tests are underway with the AVL 2000 system.

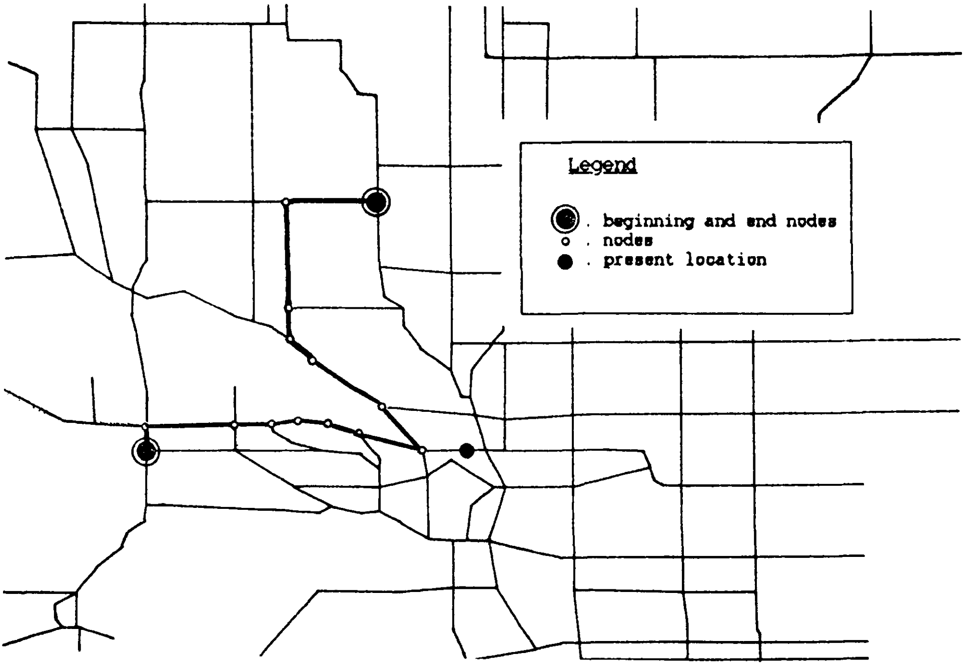


Figure 4. Electronic Map of a Municipal Polygon Containing Calgary

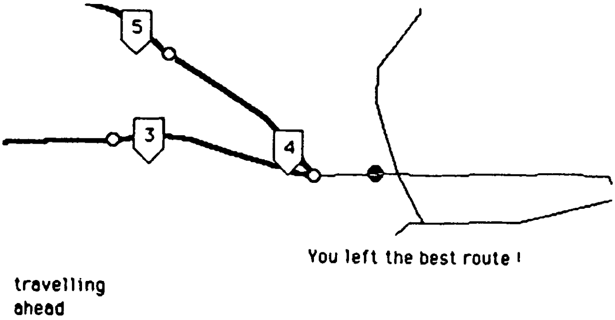


Figure 5. Zoom-View of a Portion of the Best Route

ACKNOWLEDGEMENT

This research has been carried out under an NSERC grant. Rob Goss, computer analyst in the Department of Surveying Engineering, is acknowledged for programming done in connection with producing a customized electronic map. Mona Tod typed the manuscript.

REFERENCES

Dijkstra, E.W., A Note on Two Problems on Correxiom with Graphs. Numerische Mathematik 1, 269-271 (1959).

Karimi, H.A., Krakiwsky, E.J., Harris, C., Craig, G., and Goss, R., A Relational Data Base Model for an AVL System and an Expert System for Optimal Route Selection. Eighth International Symposium on Automation in Cartography, AUTO-CARTO 8, Baltimore, Maryland, March 30-April 2 (1987).

Luenberg, D.G., Optimization by Vector Space Methods. Wiley (1969).

Skomal, E.N., Automatic Vehicle Locating Systems. Litton Educational Publishing, Inc. (1981).

Yeung, C., Alberta Transport Link-Node Network Flat File. Alberta Transportation, Edmonton (1986).

A RELATIONAL DATABASE MODEL FOR AN AVL SYSTEM AND AN EXPERT SYSTEM FOR OPTIMAL ROUTE SELECTION

Hassan A. Karimi, Edward J. Krakiwsky,
Clyde Harris, Guy Craig, and Rob Goss
Department of Surveying Engineering
The University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4

ABSTRACT

During the course of prototyping an Automatic Vehicle Location (AVL) system it was discovered that more than one database is needed, and that the map database (city, province, state ..., etc.) was conceived of being the primary database. The use of a microcomputer along with a real-time requirement are the two dominant constraints on an AVL map database which differentiates it from other types of map databases. Specifically, real-time response has an impact on the nature of database communication, and use of a microcomputer limits the primary and secondary storage capacity. After creating the infological (real-life) model it was concluded that a relational database was needed for the datalogical (computer-oriented) model. A knowledge base which is derived from the map database, is envisaged as the prime source of information for the design and implementation of an expert system for the optimal route selection.

THE UNIVERSITY OF CALGARY'S AVL SYSTEM (AVL 2000)

AVL Systems

An Automatic Vehicle Location (AVL) system is mostly referred to as an assembly of technologies and equipment that permits centralized and automatic determination, display, and control of the position and movement of a vehicle throughout an area [Skomal 1981]. A number of investigators, among which are automobile manufacturers, have already taken up the challenge to design and develop AVL systems. Although these investigations have common objectives, they are different in their approach and techniques. For example, AVL systems employ different positioning techniques (e.g., dead-reckoning, radio communication, or satellite positioning), depending upon the positioning accuracy needed, the cost, etc.

Several alternatives exist when developing an AVL system. Choosing which alternative should be used depends on the requirements. Some applications of AVL systems include emergency, fleet management, delivery trucks, transit buses, taxicabs, and couriers. For some of the applications, a dispatch is necessary; for the others it is an option, hence the requirements dictate the configuration. Clearly, one AVL system could not satisfy the requirements of all different applications, thus a separate AVL system is needed (special-purpose AVL systems).

The hardware component of an AVL system includes a positioning facility, a computer, input and output facilities, and a storage facility. The software component of an AVL system is mainly composed of a control program, databases, interfaces, and input files.

Prototype

The development of an AVL system with the extension of its application in surveying and mapping, has been initiated at The University of Calgary [Krakiwsky et al. 1987] - AVL 2000. From the start, the overall design criteria considered the present needs and the future advancement. The objective of the project was to build an in-vehicle real-time system which utilizes a GPS receiver for positioning and a digital map for position superimposition. Also, an optimal route algorithm was to be built into the software component. That is, by choosing a destination point, the best route - based on some criterion such as time, distance, ... etc. - between the source and this point is computed in real-time. To realize and solve the practical problems associated with the above objective and to pilot future trends, the AVL 2000 system concept was prototyped.

A microcomputer (Macplus with 1 megabyte RAM), a 20 megabyte hard disk, an input device (keyboard), an output device (CRT), and a GPS receiver (Trimble) make up the hardware component. The choice of the hardware components was not considered crucial as it was mostly the software component which had to be prototyped. A control program, map database, optimal route database, and input coming from the receiver comprise the software component. Some of its functions include map display, highlighting the optimal route, and indicating the moving vehicle position on the map. Figure 1 illustrates the configuration of the AVL 2000 prototype.

First and Second Generation Systems

The design of the first generation AVL 2000 system is presently being developed with specific applications in mind. The hardware component of the first generation is to be composed of a microprocessor, CD-ROM, and a voice synthesizer for both input and output. Some AI concepts will be part of the first generation software component.

Clearly, the prototype is microcomputer-based while the first generation is a microprocessor-based (a microprocessor totally devoted to the AVL tasks). The second generation system will be a VLSI-based (a VLSI chip designed to perform all the required AVL tasks).

MAP DATABASE

One of the key elements in any AVL system is the digital route map. Regardless of the application area, the

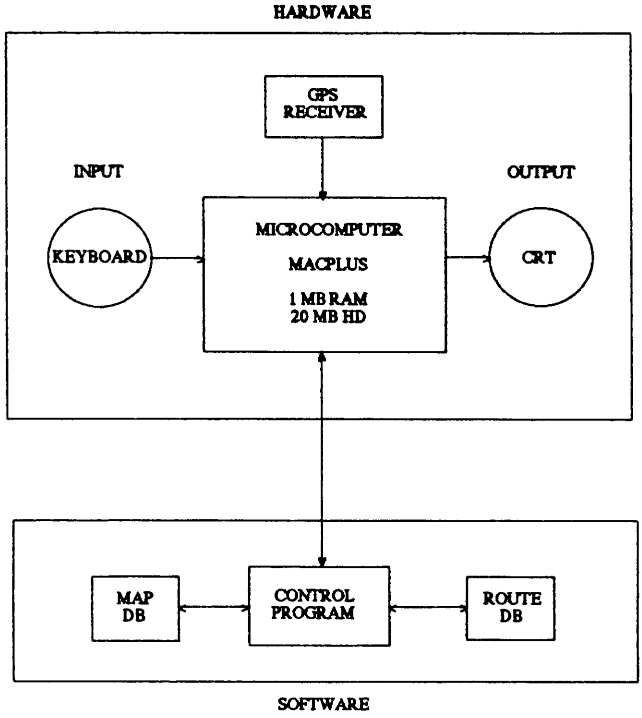


Figure 1. The AVL 2000 prototype configuration

spatial data of an area's road network along with the spatial topology organized and managed in a database are required. This map database was even needed at the very early stages of prototyping.

Requirements

Since the AVL 2000 system is a microcomputer-processor based system, has a real-time system requirement, and has limited storage, the nature of the map database is affected significantly. The requirement of being a real-time system has an impact on the response of database communication, while limited primary and secondary storage limits the quantity of spatial data in the system.

Sources

Two plausible sources of a digital street map are perceived if it is not already available in the form usable in an AVL system. The first one involves digitizing map sheets of various scales; the second one involves adaptation from any existing digital map, which may have been originally developed for other purposes. The answer to the question of which method should be used will depend on the availability of the existing digital maps. For example, the ETAK [Honey et al. 1986] database is derived from a combination of U.S. Census Bureau Dual Incidence, Matrix Encoded (DIME) files, U.S.G.S. 7-1/2 minute quadrangles, aerial photographs, and local source material as needed.

Obtaining a digital map database was perceived as being the main bottleneck during prototyping. Digitization was ruled out as it was more time consuming. A digital highway network of the province was the only suitable source in our operating area. The provincial file (a flat file) had to be transformed into the form of a database usable in the AVL 2000 system.

Storage

AVL systems to date have used three different methods of map storage: photographic, digital image and digital encoded. Honda's Gyrocator and Omni Devices' navigator both use photographic map storage [Cooke 1985]. Chrysler's Stealth uses digital images (video disk images) of paper maps in its "CLASS" (Chrysler Laser Atlas Satellite System) [Cooke 1985]. A digital encoded map is a data file describing the road networks by topology using vector format. This method is more flexible and more compact than the first two methods. The ETAK [Honey et al. 1986] navigation system uses digital encoded map storage. This method more easily provides flexibility in windowing and zooming than the other two. Also, this method is more appropriate for the computations required for optimal route selection algorithms [Cooke 1985].

In the prototype, the highway map database was digitally encoded. Some of the reasons for this include flexibility, storage compactness, map display, and selection of destination by street address or intersection. Of these, the compactness of the database storage is of particular importance. Even when digitally encoded, the quantity of spatial data is usually very large; for example, the road network of the San Francisco Bay area requires between 7 and 10 megabytes of data storage space [Zavoli et al. 1985].

Data Modelling

Once the spatial data of a road network are gathered and stored, they have to be organized and managed along with their spatial relations in the form of a database system. Different approaches for formally representing relationships in a defined data structure give rise to various data models - most notable are the hierarchical, network, and relational models [Everest 1986]. In the network data model, many-to-many relationships can be represented. The hierarchical data model, which actually is a special case of the network data model, represents one-to-many relationships. In these two data models, all relationships between entities must be explicitly defined for the physical access paths. In contrast to these two data models, the relational data model excludes physical access path information. All relationships are implicit in explicitly defined attributes in the entry types.

In this project, after some investigation, it was realized that any of these data modelling approaches could be used, but implicitly defining relationships would provide more flexibility in structuring and future updates. Consequently, the relational data model was implemented. Moreover, as this map database may be used by some other modulas, such as optimal route determination, the relational model would provide all these various views more easily and in a shorter time. The AVL real-time characteristic was considered to be the main constraint in implementing a database in the system. It was decided that a polygon-based relational database would satisfy this. In a polygon-based database, a given digital road map is divided into polygons, and each polygon is used as a reference entity. Figure 2 illustrates the AVL 2000 polygon-based concept (the entities and their relationships) for road networks. Not only will this method support the necessary real-time response (since at any time, only a small portion of a large digital map is accessed and used) but also this makes it easier to implement the optimal route determination algorithms. In this database, one relation (file) was designated as the polygon directory which contains the most immediate information such as polygon ID, number of nodes in each polygon boundary, etc. as its attributes.

Different applications need the same digital spatial data of the road map, but each would view it differently. It is suggested here that a polygon-based relational database can more easily support these different views of spatial data. This can be done by means of defining polygon boundaries for each separate application. This is to say that the original route map can be divided into polygons many times, and each time the criterion for boundaries can be related to the requirements of a different application area. Doing this in developing an AVL system will optimize the flexibility in views, real-time response, and user-friendliness of the system. On the other hand, applying the same polygon boundaries for all different applications, would allow the system to be optimized for only a few cases.

OPTIMAL ROUTE DETERMINATION

Optimal route determination is the computation of the "best" route between the source node and a selected destination node. "Best" could mean shortest, fastest, safest, etc. depending upon some criterion related to the underlying application. For example, for an emergency

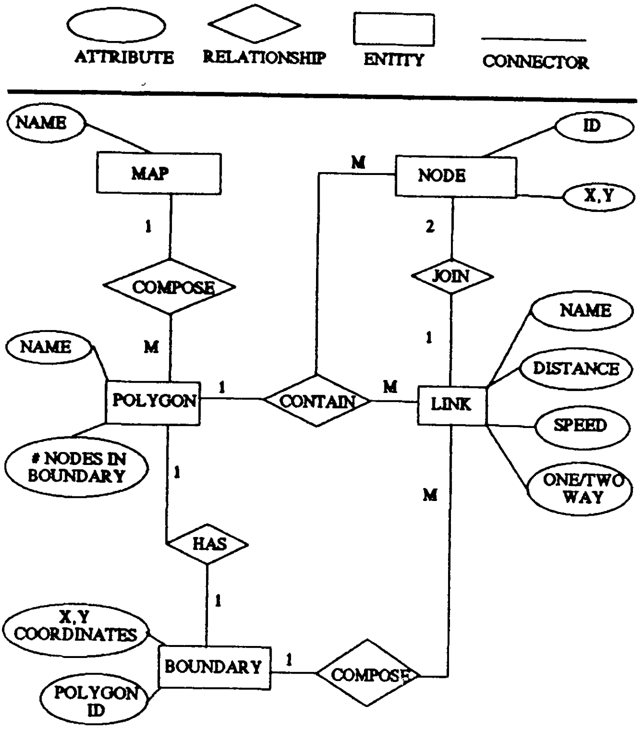


Figure 2. Elements of Entity-Relationship Model (ERM) and a geometric ERM for the AVL 2000 system.

ambulance driver, "best" usually means the fastest route, whereas for a tourist driving in a car, it could mean a route with more scenery.

Mathematically, a collection of a set of nodes and a set of links is called a network. Types of networks, each for a certain application, range from the simplest form of deterministic non-time-dependent travel times networks [e.g., Dijkstra 1959] to probabilistic time-dependent travel times networks [e.g., Hall 1986]. Algorithms have been developed in the past to determine the optimal path between any pair of nodes in such networks. These algorithms vary according to the type of network and type of application they were designed to handle. However, optimal route determination algorithms are known to be NP-complete [Sedgewick 1983].

In the AVL 2000 prototype, a deterministic non-time-dependent travel times network was assumed. Deterministic networks are networks whose links are associated with a specific known weight. Such a weight might include the length of a link or the link travel time. Non-time dependent networks are networks whose link weights do not deviate over time. For the prototype, three classes of weights were considered: (a) distance (travel length), (b) time (travel time), and (c) cost (travel cost). Of the three, only distance is an independent weight, the other two are functions of distance. For simplicity, distance (travel length) was used in the prototype. When the weight is distance, the optimal route is referred to as shortest path. Several algorithms have been developed for the standard node to node shortest path problem. Of these, Dijkstra's [1959] algorithm has been shown to be the most computationally efficient. Its computational time is proportional to n^2 ($O(n^2)$, where n is the number of nodes in the network. One of its shortcomings is that it cannot handle negative weight, but this is not considered as a problem in the AVL 2000 prototype (no negative distance).

Patch-Quilt Concept

It was mentioned that the computation time for this algorithm ($O(n^2)$) is better than some of the alternatives. However, even this is not adequate in a real-time system, especially when a large network is used. To overcome this problem, a "Patch-Quilt" concept was developed. It simply means that for those optimal route selections that require a significant amount of time, both Dijkstra's algorithm and a look-up table are used. The look-up table contains the optimal routes between any pair of polygons in the network. The look-up table entities were computed a priori using a VAX/VMS system. To compute the shortest path between any pair of polygons, the many-to-many nodes shortest-path algorithms must be used. This is because each polygon has many entrance-exit nodes. Such algorithms have been developed by a number of investigators such as Floyd [1962] and Dantzig [1966]. These algorithms are more efficient than almost any one-to-one node algorithm. This is generally true because the application of a one-to-one algorithm for a many-to-many case requires a large number of iterations.

If the destination node occurs in the same polygon as the source node, Dijkstra's algorithm will be used. This can be done in real-time without much overhead since each polygon is a fragment of a large network. On the other hand if the source and the destination nodes occur in separate polygons, the Patch-Quilt concept will be used. To accomplish this, three pieces of the final optimal route between the source and the destination are computed individually. These three are:

(a) optimal route between the pair of polygons;
(b) optimal route between the source node and the entrance-exit node marked by (a) within the source polygon; and
(c) optimal route between the destination node and the entrance-exit node marked by (a) within the destination polygon.

Optimal route between the pair of polygons is retrieved from the look-up table. This route joins one of the entrance-exit nodes of the source polygon to one of the entrance-exit nodes of the destination polygon. Then Dijkstra's algorithm is used in real-time for part (b) and (c). Finally, these three pieces will result in the required optimal route between the source and the destination.

EXPERT SYSTEMS IN AVL

During prototyping, several problem domains of AVL systems applicable to expert systems development were discovered. These include the AVL system management software, database management, database interface, and optimal route determination. In this paper, the concept of an expert system for optimal route determination is discussed, but before doing so, let us give briefly the definition of an expert system used herein.

Expert systems are computer systems that think and reason as an expert would in a particular domain, and solve real-world problems using a computer model of expert human reasoning, reaching the same conclusions that the human expert would reach if faced with a comparable problem [Weiss et al. 1984]. In contrast to a conventional program, an expert system is believed to have self-knowledge: knowledge about its own operation and structure [Hayes-Roth et al 1983]. A system often confused with an expert system is a knowledge system. Expert systems perform as human experts do whereas knowledge systems perform simpler tasks that normally require human intelligence.

In the prototype, an algorithmic approach was taken as the basis for optimal route implementation. It is seen that this technique may not be sufficient for some applications, as for the solution it only requires parameters whose values are certain and obtained a priori. Certain applications may require some other parameters that cannot be obtained a priori and whose values are time-dependent. These parameters, which can be called real-time parameters, include time of the day, season of the year, weather conditions, accident information, detours, road condition, traffic flow, visibility, and purpose of trip. Some of their values, which can be input to the system interactively, are uncertain. This would result in uncertain inferences.

Clearly, introducing real-time parameters to the system would make the task of reasoning and decision-making difficult enough to require expertise. For this, an expert system specialized in optimal route determination is developed for the first generation system. The domain-specific expert must find the best routes in a network in real-time using all real-time information. An expert who does this can be called a "route finder consultant".

The first generation system will use the algorithmic methodology blended with heuristic knowledge. This is one of the main characteristics of a true expert system. A heuristic is a rule of thumb, a trick, strategy, simplification, or other method that aids in the solution of complex problems. Heuristic generally reduces the size of the space in which one needs to search for solutions to the problem at hand.

The rules and facts associated with both the real-time parameters and the map database will be used in the construction of the system's knowledge base. A knowledge base - a collection of rules and facts - is the main source of information for the inference engine (reasoning strategy) of an expert system.

Further, for the treatment of uncertainty in this expert system, one of the several methods which have been developed to take the uncertainty into account will be considered (e.g. Bayesian method, or the concept of "fuzzy set" originated by Zadeh [1978]).

CONCLUSION

Specific application requirements should be the basis of the design and the development of an AVL system. A digital encoded map is recognized as the best map storage technique. This provides and supports many common and uncommon features required by most applications. Of the three well-known data modelling approaches, the relational model provides and supports more easily the real-time response, compact data storage, and different views of the spatial data. An algorithmic approach was taken for the optimal route determination in the AVL 2000 prototype. It was realized that this would involve a significant amount of overhead in a real-time system. For this, a "Patch-Quilt" concept using a priori polygon to polygon shortest-path computation was developed. For the first generation, it is intended to include some AI techniques and tools, such as expert systems, into the system. Among problem domains of an AVL system applicable to expert systems development, optimal route determination is of particular importance. For this, real-time parameters whose values are input interactively along with some information derived from the map database will be the basis of defining rules and facts to form a knowledge base.

ACKNOWLEDGEMENT

This research was conducted under a Natural Sciences and Engineering Research Council of Canada grant.

REFERENCES

Cooke, D.F., Vehicle Navigation Appliances. Proceedings on Digital Representations of Spatial Knowledge, AUTO-CARTO 7, Washington, D.C., March 11-14 (1985).

Dantzig, G.B., All Shortest Routes in a Graph. Operation Research House, Stanford University, Technical Report 66-3, November (1966); also in Theorie des Graphs, pp 91-92, Proceedings of the International Symposium, Rome, Italy, July (1966), Published by Dunod, Paris.

Dijkstra, E.W., A Note on Two Problems on Correxiom with Graphs, Numerische Mathematik 1, 269-271 (1959).

Everest, G.C., Database Management: Objectives, System Functions, and Administration, McGraw-Hill, Inc. (1986).

Floyd, R.W., Algorithm 97, Shortest Path, Comm. ACM 5, 345 (1962).

Hall, R.W., The Fastest Path through a Network with Random Time-Dependent Travel Times, Transportation Science, Vol. 20, No. 3, August (1980).

Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., Building Expert Systems, Addison-Wesley Publishing Company, Inc. (1983).

Honey, S.K., White, M.S. and Zavoli, W.B., Extending Low Cost Land Navigation into Systems Information Distribution and Control. IEEE Symposium on Position Location and Navigation, Las Vegas, Nevada, November 4-7 (1986).

Krakiwsky, E.J., Karimi, H.A., Harris, C. and George, J., Research into Electronic Maps and Automatic Vehicle Location. Eighth International Symposium on Automation in Cartography, AUTO-CARTO 8, Baltimore, Maryland, March 30 - April 2 (1987).

Sedgewick, R., Algorithms, Addison-Wesley Publishing Company, Inc. (1983).

Skomal, E.N., Automatic Vehicle Locating Systems, Litton Educational Publishing, Inc. (1981).

Wiess, S.M. and Kulikowski, C.A., A Practical Guide to Designing Expert Systems, Rowman & Allenheld, Totawa, N.J. (1984).

Zadeh, L.A., Fuzzy sets as a basis for a theory for possibility, Fuzzy sets and systems, New York, North-Holland (1978).

Zavolli, W.B., Honey, S.K. and White, M.S., A land vehicle naviqation system supported by digital map data base: ADPA Artificial Intelligence and Robotics Symposium, November 6-7 (1985).

The Map-Environment Interface: Surrogate Travel by Videodisc

Laurence W. Carstensen and Allan B. Cox
Virginia Polytechnic Institute and State University
Department of Geography
Blacksburg, VA 24061

The videodisc is an exciting advance in the realm of data storage. Capable of storing up to 108,000 still frame images on a single disk, the videodisc allows random access to each image, from each image, in under three seconds. As such, a new field known as videodisc surrogate travel has evolved in which users are in control of their movements through a simulated or geographic environment. Views of the surroundings are displayed on a color monitor by the videodisc. Within cartography, various projects have begun to use existing discs, and occasionally to create disks as map storage devices. This project deals with a computer aided instructional package with videodisc and microcomputer to teach map reading skills. A system is being designed in which a student must use sets of still photographs to locate his position on a map, and to move through the environment and complete a prescribed course in a minimal amount of time. This simulation of the sport of orienteering is a valuable resource in teaching map-land relationships, and in sharpening visual skills such as inspection and resection. In addition, the system is a valuable research tool for studying the methods by which these skills are learned.

THE BBC DOMESDAY SYSTEM: A NATION-WIDE GIS FOR $4448

David Rhind
Birkbeck College, University of London,
Malet Street, London WC1E 7HX

and

Stan Openshaw
University of Newcastle,
Newcastle, NE1 7RU.

ABSTRACT

This paper describes selected aspects of the Domesday project led by the BBC and intended to create an exhibition of Britain in the 1980s. Among other things, this resulted in what is perhaps the first example of a second-generation GIS. Based upon a microcomputer linked to a new LV-ROM, this holds 54000 images (maps, photos, satellite images, etc.), 300 mb of digital data and millions of words of text per side of video disk; the different types of data are cross-referenced by geographical position or by theme. Access to the data is by pointing at maps, by specification of place name or geographical coordinates or through use of a thesaurus: the source and storage form of the data is transparent to the user. Included in the initial disks are 21000 files of spatial data showing national coverage down, in some cases, to 1km square resolution. Data sets stored include geology, soils, geochemistry, population, employment and unemployment, agricultural production and land use/land cover. Though the normal purchase price of the data held would be over $400,000,the price charged to schools for the complete system - hardware, software and data - at the launch date in November 1986 was $4448. All this has significant implications for the spread of use of geographical data bases and GIS technology.

INTRODUCTION

In 1086, a comprehensive survey of much of England was carried out at the behest of William the Conquerer. Nine hundred years later, a collaborative project led by the BBC has repeated the exercise and extended its scope to cover all of Britain and to include an enormously increased range of information. The results of this $5 million project include:

- a micro-computer system, part of which is a GIS
- a new video disk player, capable of storing and overlaying information held in both analogue and digital form
- two video disks containing 30 million words, 21000 spatial (or mappable) digital data sets, 24000 Ordnance Survey topographic maps, statistical tabulations and time series, picture libraries and TV film clips

The project effectively began in December 1984, the launch of the final product being 23 months later. In that time, the development of data storage and access concepts, the design and construction of the hardware and of the software, negotiations to obtain data sets from government, private sector,academic and other agencies, the organisation of 14000schools to collect certain types of information, the validation of the data and the construction of documentation were all completed. Details of the project and of the organisation of the various teams working in different parts of Britain are, however, not the concern of this paper (see Goddard and Armstrong 1986). Equally, the voluminous non-geographical data, both analogue and digital, are not relevant here though they include numerous libraries of photographs on topics as diverse as the Royal Family and British Design Council Award winners, ceramics and public houses(pubs). We concentrate on the spatially-related facilities which the Domesday machine makes available and claim that, inter alia, it can be considered the first example of a second generation GIS. Our justification for this is as follows:

(i) it handles data in both analogue and digital form and permits graphic overlay of one on the other, plus some digital operations in relation to analogue maps

(ii) it comes complete with its own data base, currently totalling in excess of 500 megabytes and covering a vast range of environmental ,demographic, socio-economic and other variables

(iii) it provides high response rate interactive graphics

(iv) it implements the cross-linkage of maps, air photographs, colour slides, moving pictures, text (held digitally) and digital numerical data. Thus a user interested in one geographical area can move from one type of information to another virtually instantaneously

(v) it is extremely easy to use: successful demonstrations to the Prime Minister were given by 11 year old school children who had only two hours practice on the system

(vi) it is very cheap: schools can purchase the entire hardware, software and data for $4448 (based on January 1, 1987 exchange rates) whilst other purchasers pay $5930 plus $890 tax. Contrast this with the normal purchase price of those digital data sets on the Domesday disks which are readily available: their cost would exceed $400,000

(vii) despite (v) and (vi), it includes several desirable capabilities which are not commonplace: it permits the user, for instance,to study the effects of changing the data resolution along various scale hierarchies (e.g.electoral wards to administrative districts to counties to regions or 1,2,3... 10 km squares)

We go on to describe the computer system and the data base in more detail before concluding with a consideration of the likely effects of the advent of the Domesday machine.

THE HARDWARE AND SOFTWARE

The initial release of the system consists of:

- a BBC 128k Master Series micro-computer, including floppy disk drives and tracker-ball
- a new Philips Laservision 12 inch LV-ROM
- a high resolution colour monitor
- retrieval and analysis software
- two Domesday disks, the national and the local (or 'Community') disks

Openshaw, Wymer and Charlton (1986) have described the basis of the system in some detail. Appendix 1 provides brief technical details of the initial hardware. In fact, a rather more powerful micro (the RML Nimbus) is also now available to drive the system whilst an IBM PC - compatible version is scheduled to be available by February 1987.

Extensive use is made of the tracker-ball to access the data, by pointing at items in menus, at positions on maps or at keys in statistical displays. The default access to the national disk, for instance, is by navigating a picture gallery, each picture representing a topic which may be pursued; alternatively, the user may 'walk'out one of the doors into different types of environment. In addition, keywords, or keyword strings,place names, National Grid Reference coordinates and other items may be entered in the normal way via the keyboard.

The software is written in BCPL and was produced by Logica Limited under contract to the BBC. The data structure utilised for storing the spatial ('mappable') data files was devised by one of the authors (SO) and colleagues and tested initially on a VAX computer. In essence, all data on the system are held in raster form because of memory limitations in the initial microcomputer. Thus attribute data for administrative areas are stored as fixed length lists and the vector boundaries of the areas are held as a highly compacted equivalent with pointers between the two data sets. Default values for class intervals and various other characteristics of each of the 21,000 data sets of this type were computed and stored at load-time.

THE DATA BASE

Four main contractors were charged with obtaining and/or reorganising data for the Domesday disks. These were the Birkbeck College Department of Geography, the Centre for Urban and Regional Development Studies at the University of Newcastle, the Economic and Social Research Council Data Archive at Essex University and the Institute of Terrestrial Ecology, Bangor. The different teams had usually disparate but sometimes overlapping responsibilities: as a consequence, collaboration was essential and constant use was made of the UK Joint Academic Network (JANET) for electronic mail and for the transferring of certain data sets.

The initial Domesday 'package' includes two video disks. We now consider selected aspects of each in turn.

The local disk

This is a 'peoples data base' on Britain, in so far as it was compiled by nearly one million individuals and represents the aggregate of their views on small areas of the country. The bulk of information on this disk consists of Ordnance Survey (OS) topographic maps, including complete coverage of the country based upon 1/50 000, 1/250 000 and smaller scape maps and 1/10 000 scale maps for 80 cities; larger scale maps, floor plans etc of sites of special interest; Landsat Thematic Mapper and Multi-Spectral Scanner imagery ; 30 million words of descriptive text; colour slides of locations throughout the country; and the OS gazetteer. The information is arranged hierarchically as shown in table 1; 'zooming in' and out between these levels, scrolling across country and changing from text to pictures to topographic maps for the same area is virtually instantaneous. Moreover, digital operations on the analogue maps, such as measuring length and area, are possible.

The level 0 and 1 essays were written by academic geographers, whilst those at level 2 were written by professionals, ranging from school teachers to unversity professors. The level 3 text is extremely heterogeneous, being written in many cases by school children and is available for about 9000 4 x 3 km areas in Britain.

The national disk

The data on the national disk are of three main types: a set of picture libraries showing many aspects of British life in the 1980s; several thousand cross-tabulations of statistical data derived from government series such as the Family Expenditure Survey; and also a variety of spatial data. Table 2 illustrates the highest resolution spatial data sets held which pertain directly to land or to people. Numerous other data sets exist on the disk, some of which are documented by Owen, Green and Coombes (1986). In general, the data are held and are available by whatever 'standard' geographical areas were used to report them: these areas include grid squares, parliamentary constituencies, districts, counties, functional

TABLE 1

Level	*Area name/size*	*Data held*
0	UK	Essay, Landsat MSS mosaic
1	Regions (N & S Britain, N. Ireland, Isle of Man, Orkney and Shetlands, Channel Islands)	Essays, Landsat MSS mosaics
2	40 x 30 km blocks (covering c 70% of the UK)	Essays, Landsat TM true colour and false colour images and up to 5 air photographs
3	4 × 3 km blocks (covering c 45% of the UK)	Essays, up to 3 colour slides

Key TM Thematic Mapper MSS Multi-Spectral Scanner

TABLE 2

Data	*Source*	*Resolution (x × x km)*	*Coverage*	*Location*
Place names	OS	1	UK	C
Physiography and topography				
(maps)	OSGB and OSNI	varies with scale	GB and NI	C and N
(data)	ITE	10	UK	N
(text)	schools etc	various	45–70% GB	C
Land use cover	schools etc	1	45% GB	N
Incidence counts	schools etc	1	50% GB	N
Solid geology	BGS	1	GB	N
Drift geology	ITE	10	UK	N
Soils	Soil Surveys of			
	England and Wales	5	E & W	N
	and of Scotland	5	S	N
	ITE	10	UK	N
Geochemistry	AGRG	5	E & W	N
Climate	ITE	10	UK	N
Water features	ITE	10	UK	N
Woodland	ITE	10	GB	N
Fauna and flora	ITE	10	UK	N
Agriculture	MAFF	5	E & W	N
	DAFS	5	Scot	N
	MANI	10	N. Ire	N
	ITE	10	GB	N
Land quality	ITE	10	GB	N
Pollution	various	varies	GB	N
Population	OPCS/Birkbeck	1	GB	N

Key BGS British Geological Survey, ITE Institute of Terrestrial Ecology National Environment Research Council AGRG Applied Geochemistry Research Group Imperial College MAFF Ministry of Agriculture Fisheries and Foods DAFS Department of Agriculture and Fisheries for Scotland MANI Ministry of Agriculture for Northern Ireland, 'Primary' spatial data set indicates the highest resolution data set of that variable in many cases the ITE data bank provides additional data at lower resolution C Community disc N National disc

Source: Rhind and Mounsey (1986)

regions, Local Education Authority areas, televisıon regions and Regional Health Authority areas. Thirty three dıfferent sets of areas were used in total. Many sets are held at multiple levels of geographical aggregatıon.

It should be emphasised that the data listed in table 2 are only the so-called 'spatial' data i.e. those which can be mapped and (gıven suitable software) manipulated on a spatial basis. Many other environmental data sets exıst on the dısk which can be tabulated and most of these 'non-spatial' data sets are crudely classified by geographical area: forestry data, for ınstance are only available in 'non-spatial form' but include details of the areal extents of different species of woodland trees for each county ın Britaın.

In addition to the variables drawn from offıcıal data sources, a schools-based proȷect led to the collection of nearly 70 items of data for each of over 100,000 1 km grid squares. Nearly half the schools in Brıtain took part in this data gathering exercise in which prımary, secondary and tertiary land use and land cover was recorded for each 1km square, together with the number of occurrences of facilıties such as banks, leisure centres and schools (Rhind and Mounsey 1986).

Some aspects of the compilation of the national disk data were novel. Perhaps the best example of this ıs Green's creations of a 1km grid data set of Population Census data for the whole of maınland Brıtain. He took the data for 125,000 small but irregularly sızed Enumeration Dıstricts, spatially described only by a coordinate pair locating the centroıd of the area; from this, he generated a Dirıchlet tesselatıon, clipped thıs by superimposing a detailed coastlıne of the whole country, rasterısed these Dirichlet tıles to 100 m resolution - producing, ın effect, a 12000 by 8000 matrix of ED names - recombined these into 1km areas by allocating the appropriate fraction of each ED's populatıon to the larger grid cell and restored the unpopulated areas known to exist by a thresholdıng process. The entıre operatıon ıs described in more detail in Rhind and Mounsey (1986 p.323). In additıon, the requirement to have clear video images of background maps to underlay statistıcal or thematic maps ensured that some 1500 simple monochrome plots had to be generated from OS digital 1/625000 scale map data, photographed and indexed.

Access to the national data is primarily through a hierarchical thesaurus. From the uppermost level of four topics (Culture, Economy, Environment and Society), a 7 level cross-lınked structure expands to gıve over 9,000 basic terms by whıch access is gained to text, pictures or digıtal data.

Functıonality

The capabılıties of the system so far as spatıal data are concerned include the ability to:

(ı) view topographic maps, to scroll across the entire country and to 'zoom in' from national to very local views by moving from small scale to large scale maps of the same area

(ii) measure area and distance in metric or imperial units by indicating boundaries or routes on the video topographic maps

(iii) view satellite imagery, air photographs, slides and text as well as maps; to store relationships between these entities and retrieve in accordance with the relationships

(iv) retrieve data by area name, by coordinate position, by predefined geographical 'window' and/or by variable

(v) plot digital spatial data with either default or user-specified class intervals, colour schemes, areal limits, data resolution etc.

(vi) overlay these plots, if desired, on background topographic maps

(vii) interrogate the display to obtain the value (or area name) at a point of interest

(viii) compute the statistical correspondence between variables in the geographical window selected

(ix) dump selected data from the video disk onto floppy disk and to incorporate in a display the user's own data supplied on floppy disk with that from the LV ROM

(x) leave an 'audit trail' where the user has 'been' in the system so that bookmarks may be created to guide others directly to items of interest when in a teaching environment. Additional capabilities which are more relevant with the other data sets include the ability to display time series data (e.g. newspaper sales by region by time) as moving images and to provide surrogate walks (in which the user 'walks' around a farm, a town, etc. progressing and turning around at will and zooming in to examine items of particular interest).

DOMESDAY AND THE FUTURE

It is clear from the above description that the initial Domesday system is revolutionary. Nonetheless, it still has some shortcomings. The most serious of these are:

- the limited analytical capability of the present software. This stems from two factors: the time available to write reliable basic, let alone sophisticated spatial analytic, software and the memory constraint imposed by the use of the BBC microcomputer. Clearly, the advent of an IBM PC version will reduce both problems over the next few months. In the medium term, the move from 8 bit processor to 32 bit processors will transform the analytical capabilities and will further extend the graphics interface split screen working, etc.
- the lack of any regular up-dates for those data which are needed in highly topical form. Plans have already been laid for regular up-dating and publishing of new video disks, subject to sufficient user demand.
- the limitations of the data base to Britain. Over a dozen other countries have already expressed strong interest in replicating the Domesday project; within Britain, compilation of a new rural heritage disk is about to begin and several others (e.g. on London) are planned to follow.
- the possible misuse of data in combination through analyses carried out by unskilled users. It is evident that certain combinations (altitude with % unemployed) are probably meaningless; more seriously

combining data derived from, say, maps at widely different scales may be most misleading. In the longer term, the only solution to this problem - encountered by all GIS - is to install a suitable expert system front-end processor. In the short term, human guidance and education is the only solution.

The Domesday Project has already had a major impact in Britain in the way in which it has removed some data from the private domain of data gatherers into the public domain. In some cases, data gatherers have been loathe to lose control over their data, notably where these data ensure a steady cash return from copyright revenues.

Despite the present shortcomings, we believe the Domesday machine to be of critical importance for three reasons. The first of these stems from its ability to function with analogue as well as digital data: it is totally unreasonable to expect all historical and contemporary data to be converted into computer form before use. The second reason is that, because of its creation by an information-oriented organisation rather than by geographers or surveyors, it treats GIS capabilities as just one - though important - set of data base operations: as a consequence, the long-fostered artificial separation between GIS-type data and other data is demonstrated to be chimerical. Finally, because of these two factors and because of its exceptionally low price, Domesday and its successors may well become the most widely used information system in the world. As such, it will make much information equally available to the government planner, the commercial sector developer and the members of the lay public; indeed, it could well have an important and positive societal role in the information society.

ACKNOWLEDGEMENTS

The authors were only two members of a large collection of individuals working on this project. We readily acknowledge that most of what is reported here was a team achievement.

REFERENCES

Goddard J.B. and Armstrong P (1986). The 1986 Domesday Project. Trans.Inst. Brit. Geogr. NS 11,3,290-5.

Openshaw S., Wymer C and Charlton M. (1986). A geographical information and mapping system for the BBC Domesday optical disks. Trans.Inst. British Geogr. NS 11, 3, 296-304.

Owen D.W., Green A.E. and Coombes M.G. (1986). Using the social and economic data on the BBC Domesday disk. Trans.Inst. Brit.Geogr. NS 11, 3, 305-14.

Rhind D.W. and Mounsey H.M. (1986). The land and people of Britain: a Domesday record *Trans.Inst.Brit.Geogr.* NS 11,3, 315-26.

Appendix 1 Summary of technical specifications of the Domesday hardware

BBC Master Advanced Interactive Video (AIV) Microcomputer. A specialised derivative of the standard Master series micro incorporating:

- 128 kb memory plus 128kb of ROM
- operating system with extended graphics
- interfaces for disk, cassette, parallel printers, serial RS232, user port, 1 MHz bus, analogue, RGB, video
- the Turbo Co processor 65C102
- internal SCSI interface
- video filing system ROM, including support for the tracker ball and display cursor

BBC/Philips AIV VP415 Laser Vision player

Front loading player incorporating a semi-conductor laser, electronic time-base corrector with sync inserter, RGB output,RGB graphics overlay, LV-ROM decoder and RS 232 interface

The LV-ROM format allows up to 324 Mb of diigtal data to be stored on each side of each disk (read only), as well as 54 000 analogue video frames. Data may be replaced with analogue audio where required, allowing either video/data or video/audio to exist simultaneously at any point

Colour monitor

14 inch 600 lines monitor with 0.42 mm dot pitch etched tube and amplifier/loudspeaker. Inputs can include CVBS, linear, RGB, TTL RGB aud audio.

ACQUIRING APPROXIMATE REPRESENTATIONS OF SOME SPATIAL RELATIONS

Vincent B. Robinson
(Goss.Ensuadmin@UNCA-MULTICS.MAILNET)
Department of Surveying Engineering
The University of Calgary
2500 University Drive, NW
Calgary, Alberta T2N 1N4
CANADA

Richard Wong
(Rnwhc@Cunyvm.BITNET)
Department of Computer Science
Hunter College - CUNY
695 Park Avenue
New York, NY 10021
USA

ABSTRACT

This paper reports on development of a Spatial Relations Acquisition Station (SRAS) and preliminary results of man-machine interactions. SRAS is based on a process for acquiring linguistic concepts through mixed-initiative man-machine interactions. It is unique by virtue of acquiring fuzzy representations while requiring only "yes/no" answers from the user. It is shown how SRAS can be used to acquire multiperson concepts for use in an organizational context. Results show significant interuser and interterm variation, and suggest that the size of the spatial database may not influence extent of interuser semantic variation. Results of multiperson concept formation show the importance of understanding the process.

INTRODUCTION

Robinson and Frank (1985) and Robinson et al (1985b) have identified several major areas where an understanding of NL concepts can contribute to our understanding of the nature and influence of uncertainty in geographic information processing. Like Robinson (1986), we are concerned primarily with the representation of NL concepts for use in the retrieval of geographic information from geographic data bases.

There have been several attempts at formulating spatial query languages. A number of spatial information query languages have been developed that are similar to Chamberlin and Boyce's (1974) SEQUEL (e.g., Frank, 1982; Barrera and Buchmann, 1981). They have much in common with other systems like the Map Analysis Package (Tomlin and Tomlin, 1981) that uses a subset of the English language to pass commands to a spatial information system. The meaning of each command must be unambiguous. Thus, the system cannot exploit the vagueness inherent in a natural language expression. One of the motivations behind development of a SEQUEL-like query language was the general lack of knowledge concerning retrieval of spatial information using natural language concepts (Frank 1982).

This paper reports on an effort to develop representations of NL concepts that preserve their approximate nature. Presented here is an algorithm for acquiring linguistic concepts through mixed-initiative, man-machine interactions operating on a spatial data base. Finally, selected results of man-machine interactions are presented and discussed.

REPRESENTATION FRAMEWORK

It is within the context of this framework that the approximation of linguistic entities is developed. Emphasis is placed on PRUF and test-score semantics, both of which have a basis in fuzzy set and possibility theory (Zadeh, 1978, 1981). PRUF and test-score semantics provide a general meaning representation and translation framework for development of linguistic approximations of spatial concepts for retrieval of spatial information from a spatial data base.

Briefly, Possibilistic Relational Uniform Fuzzy (PRUF) meaning representation language is based on the assumption that the imprecision intrinsic in natural languages is possibilistic in nature. Hence the logic underlying PRUF is a Fuzzy Logic. In PRUF a relational database is a collection of fuzzy relations which may be characterized in various ways by tables, predicates, recognition algorithms, generation algorithms, etc. Since an expression in PRUF is a procedure, it involves, only the frames not the relations in the database.

The semantics underlying PRUF are test-score semantics (TSS). The basic idea underlying TSS is that an entity in linguistic discourse has the effect of inducing elastic constraints on a set of objects or relations. The meaning of the entity may be defined by (a) identifying the constraints that are induced by the entity; (b) describing the tests that must be performed to ascertain the degree to which each constraint is satisfied; and (c) specifying the manner in which the partial test scores are to be aggregated to yield an overall test score.

Zadeh (1981) contends that the meaning of a linguistic entity in a natural language may be identified by testing elastic constraints that are implicit or explicit in the entity in question. The testing of constraints can be accomplished using tools afforded by test-score semantics and fuzzy logic can be used to assess the compatibility of a linguistic summary with a given database. The process of meaning representation in test-score semantics involves three distinct phases - (1) an explanatory database frame or EDF is constructed; (2) a test procedure is constructed which acts on relations in the explanatory data base (EDB) and yields test scores which represent the degree to which elastic constraints induced by the constituents of the semantic entity are satisfied; and (3) the partial test scores are aggregated into an overall test score that is a vector serving as measure of compatibility of the semantic entity with the EDB.

In PRUF, the translation of a proposition may be either focused or unfocused. The focused translation generally leads to a possibility assignment equation. An unfocused translation based on TSS is a collection of tests that are performed on a database induced by the proposition; and a set of rules for aggregating the partial test scores into an overall test score that represents the compatibility of the given proposition with the database. Robinson (1986a) provides examples of these translations and their relation to this problem.

ACQUISITION OF LINGUISTIC APPROXIMATIONS

Jain (1980), Hersh et al. (1979), Leung (1982), and Lundberg (1982) have all bemoaned the lack of a clear methodology for determining compatibility functions. One of the problems has been the inability to discriminate between measuring the ability of subjects to use fuzzy logic as opposed to specifying the membership functions (e.g., Lundberg, 1982). This assumption underlies Yager's (1982) document retrieval system based on fuzzy logic and places an impossible cognitive load upon the user. The methodology outlined here reduces cognitive load while capturing the vagueness inherent in natural language concepts.

Using a mixed-initiative methodology similar to that suggested by Nakamura and Iwai(1982), compatibility functions are acquired by the Spatial Relations Acquisition Station (SRAS) (Robinson, 1984, 1986c). The process described below is designed for the problem of determining the meaning of a spatial relation such as NEAR using some base variable such as distance. It is composed of four major components. First, the process is initialized. Second, a question is chosen according to some criteria. Third, response to a question is used to infer adjustments to the representation of the linguistic variable. Finally, before repeating the second and third steps above a decision is made as to whether or not the question-answer process should continue.

In this question-answering scheme it is assumed that XA is the computer's universe of discourse on a base variable and XB the user's universe of discourse. The concept to be learned is denoted as C. The computer learns C through a process of question and answer (QA) by constructing fuzzy set FS, the learned concept, which is a replica of C in XA. The computer selects a question unit (QU) out of units of XA based on a selection criteria and asks the user whether x[i] belongs to C (x[i]εC) or not (x $\notin$ C). The user answers YES or NO.

Initialization

This process begins from a position of maximum uncertainty. This is tantamount to the machine possessing no preconception. We use the definition of maximum uncertainty as derived from the fuzzy information-theoretic measure described by De Luca and Termini (1972). In essence, each tuple in the relation receives a compatibility score of 0.5. Since this measure plays another important role in the process, it is described later in more detail.

Concept Formation Process

Let FS[k-1] denote the learned concept of the computer just before the k-th QA step. When reply of the user to the k-th question of whether x[i] belongs to set C is YES, the computer constructs FS(x) in its knowledge space XA. As a result of the k-th QA step, learned concept C[k-1] is changed to C[k] given by

$$C_k = C_{k-1} \cup FS(x). \qquad (1)$$

When the reply is NO, the computer constructs nFS(x) in knowledge space XA and constructs C[k] from C[k-1] using nFS(x) (1-FS(x)) where

$$C_k = C_{k-1} \cap nFS(x). \qquad (2)$$

As an initial approximation, FS(x) and nFS(x) are assumed to be defined as

$$FS(x) = \exp(-\alpha d_{ik}) \quad \text{where} \quad \alpha > 0 , \quad \text{and} \qquad (3)$$

$$nFS(x) = 1 - FS(x) \qquad (4)$$

where d[ik] is the distance of x[i] from x[k].

In the above specification the parameter α determines the spread of both functions. The parameter α is adaptive. Initially

$$\alpha = \ln(0.5) / \max(d_{ik}) \qquad (5)$$

which means that those locations farthest from location k would be assigned a membership value of 0.5 and those in between will range from 0.5 through 1.0 depending upon the value of the base variable, distance.

During the acquisition process α is adaptively changed. First, α [k] is changed according to the following rules -

if the answer to x[k] is NO
then $\alpha[k] = \ln(0.2) / x[k]$ (6a)

if the answer to x[k] is YES
then $\alpha[k] = \ln(0.85) / x[k]$. (6b)

These rules have the effect of drawing an analogy between the 'nearness' or 'not nearness' of x[k] to the key location and those locations similar distances from x[k]. In Eq. 6b, if x[k] is closer than the previous YES-related x[k] then the previous value is used. Thus, the spread of membership function (see Eq. 3) resulting from an affirmative response is never constricted.

To allow for previous answers let x[i] be the k-th QU and x[j] is a QU used before the k-th question-answering step. If x[j] exits in the neighborhood of x[i] in knowledge space XA, and the reply to x[i] is opposite to x[j] then it is supposed that the boundary of user's concept exists between x[i] and x[j]. Thus, the value of α is increased; that is FS(x) is made narrow so FS(x[j]) becomes below (or above) a prescribed value ξ(or $1-\xi$). Before the next QA step α is set to $\alpha[k]$.

Selection of Questions

Each place other than z is considered to be a candidate as a question unit. Candidates for the k-th question unit are limited to only the units whose grade of membership are above a prescribed small value (e.g. 0.1), which have not been used as the question units before, and have not been in close proximity to another question unit (ie receiving a membership value greater than or equal to 0.80).

It is desirable that QU's be selected so as to conform learned concept C[k] to the user's concept of interest C in knowledge space XB. In order to decide on the appropriate QU,a measure of the uncertainty of a fuzzy concept is used. An information-theoretic measure (De Luca and Termini, 1972) is used to measure the uncertainty of a fuzzy relation. It is defined as -

$$I_k = - \sum_{i=1}^{n} [(\mu_i \ln \mu_i) + ((1 - \mu_i) \ln (1 - \mu_i))]. \qquad (7)$$

This measure takes on the value of zero(0) if and only if $\mu[i] = 0$ or $\mu[i] = 1$ for all i. It is maximized when $\mu[i] = 0.5$ for all i. This latter condition occurs when the dominant truth value of any tuple cannot be distinguished.

Let Iyes(k) and Ino(k) be the measures of uncertainty of the fuzzy sets given by (1) and (2) respectively. Iyes(k) corresponds to the uncertainty of concept C[k] in the k-th QA step if the user replies YES. Ino(k) corresponds to the uncertainty associated with concept C[k] in the k-th QA step if the user replies NO. So, let

E(I(k)) be the expectation of I(k) which is given by -

$$E(I_k) = [\nu I_{yes(k)} + \omega I_{no(k)}] \tag{8}$$

where ν and ω are weights that may be used to reflect the relative likelihood a response will be YES or NO. In this process

$$\nu = \exp[-2 \alpha d_{ik}] \quad \text{and} \quad \omega = 1.0 - \nu \tag{9}$$

which has to effect of weighting the average expected uncertainty as function of the distance from the key location. This is more consistent with the manner users weight their expectations than the simple averaging as reported in Robinson et al (1985a, 1985b).

E(I[k]) is calculated only with respect to candidate units. Thus, the optimal question unit for the k-th QA step is that which

$$\text{maximizes} \quad | I_{k-1} - E(I_k) | . \tag{10}$$

As a result of simulations this was found to provide better boundary finding behavior than that described in Robinson et al (1985a).

Stopping the Process

One of the major issues in specifying this process is that of when does the process stop. Here use is made of Kaufmann's index of fuzziness (K).

Index of Fuzziness. The index of fuzziness suggested by Kaufmann (1975) is defined as

$$K_k = \min_S \; (2/[XA\#]^{\frac{1}{2}}) \; [\sum_{x \, \varepsilon \, XA} (\mu_{C_k}(x_i) - \mu_S(x_i))^2]^{\frac{1}{2}} \tag{11}$$

where S is any ordinary subset in XA, XA# is the number of units in XA, μ[Ck](x[i]) = membership function of fuzzy set C[k] and μ[S](x[i]) is the characteristic function of ordinary set S. K[k] is a normalized distance in XA# dimensional space between C[k] and ordinary set SC[k] (ε {S}) nearest to C[k], and does not become over 1.

Now consider that d(x[i]) is the projection of an Euclidean Distance between C[k] and SC[k] into the i axis :

$$d(x_i) = \mu_{C_k}(x_i) - \mu_{SC_k}(x_i) \quad . \qquad (12)$$

The value of d(x[i]) is a measure of fuzziness of x[i] with respect to membership in C[k] . The definition of SC[k] -

$$\mu_{SC_k}(x_i) = 0 \text{ for } \mu_{C_k}(x_i) < 0.5 \quad \text{and} \qquad (13a)$$

$$\mu_{SC_k}(x_i) = 1 \text{ for } \mu_{C_k}(x_i) \geq 0.5 \qquad (13b)$$

Thus, K[k] indicates how strongly correlated the fuzzy subset representation is with a crisp, or regular, subset representation. The index gives us an indication of how closely the fuzzy concept fits a 'crisp', or nonfuzzy, representation. When K = 0 then there no longer exists a difference between the fuzzy concept and the crisp concept.

Stopping Rule. In this work it is assumed that the computer has accomplished learning the user's subject of interest C when the index of fuzziness of C[k] falls under a prescribed value, say K[ke]. The computer finishes when the K[k] becomes less than some specified proportion (ρ) of the maximum of the values in the previous steps. That is to say that the process stops at step k[e] where it is the step satisfying for the first time the following relation -

$$K_{k_e} < \rho \, [\max_{k=0,1,\ldots,k_e} K_k] . \qquad (14)$$

In the case of FAR relations the algorithm remains substantially the same only making use of the complement of the results. That is to say that when the user responds with a YES to the question "is city_z FAR from city_x" the process above treats it as a negative response. Upon modification of the concept the complement is used as the representation of FAR. This has the nice property of representing the complement of NEAR if the responses of the users are consistent with FAR being strictly a complement of NEAR.

MULTIPERSON CONCEPT FORMATION

The procedures used in SRAS essentially acquire a personal definition of a spatial relation. It has meaning with regard to the semantics of the single user. However, geographic information systems are typically used within organizations that arrive at definitions by committee. We suggest here that the concepts acquired by SRAS can subsequently be used in arriving at a consensual representation of a spatial relation. Furthermore, the process by which the concensual representation is arrived at can clearly and rigorously defined.

There are several approaches to constructing multiperson concepts. We draw upon the work of Gaglio et al (1985) and discuss four methods of constructing multiperson concepts from SRAS. They are the agreement, global evidence, combined agreement and global evidence, Zimmermann's (1983) and the weighted-mean method.

Agreement Method

Fuzzy intersection of μ[i]'s defined as

$$\mu(x_k) = \bigcap_i \mu_i(x_k) \qquad (15)$$

forms the basis of this method. This corresponds to a group decision procedure where decisionmakers have a sort of veto power. That is, the degree of acceptance assigned to each truth value is equal to the lowest among those assigned by the various committee members.

Global Evidence Method

In this method the "positive" opinions prevail because it is based on fuzzy union of μ[i]'s

$$\mu(x_k) = \bigcup_i \mu_i(x_k) \qquad (16)$$

Gaglio et al (1985) suggest that this method may be suitable when the procedure for obtaining multiperson concepts does not have a feature similar to veto power of some member.

Combined Agreement and Global Evidence Methods

There are several possible ways of combining the previous two methods. We discuss two that are particularly relevant to the kind of committees typical of the organizational context of geographic information systems.

Method I. Committee member i has veto power over decisions of others and is defined

$$\mu(x_k) = \mu_i(x_k) \cap \left(\bigcup_{j \neq 1} \mu_j(x_k) \right). \qquad (17)$$

Method II. The second method defined as

$$\mu(x_k) = \mu_i(x_k) \cup \left(\bigcap_{j \neq 1} \mu_j(x_k) \right) \qquad (18)$$

corresponds to the situation where committee member i only has "acceptance" power not "veto" power. Both (17) and (18) describe situations where there is asymmetric decisionmaking power among the committee members. Asymmetry in committee situations generally is a function of the type of chair a committee has. Therefore, we suggest that (17) is a reasonable method for modelling a "strongly chaired committee" while (18) can be used to model a "weakly chaired committee".

Zimmermann's Method

In the method suggested by Zimmermann (1983) we combine the agreement with the global evidence method using a "compensatory and" operator. This method may be defined as

$$\mu(x_k) = [\bigcup_i \mu_i(x_k)]^{1-\gamma} \quad [\bigcap_i \mu_i(x_k)]^{\gamma} \qquad (19)$$

where $0 \leq \gamma \leq 1$.

This method preserves symmetry among decisionmakers while striking a compromise between the global evidence and the agreement methods. The parameter γ determines the nature of that compromise. As the value of γ increases the greater the influence of the global evidence method increases.

Weighted-Mean Method

The weighted-mean method can defined by

$$\mu(x_k) = \sum_i w_i \mu_i(x_k) \qquad (20a)$$

where

$$\sum_i w_i = 1. \qquad (20b)$$

In this method the w[i]'s can be used to weight the importance of each committee member's concept. However, we see many problems with trying to formalize the specification of the w[i]'s, thus use of this method should be used only after some additional research has been conducted in this arena. In addition, use of some of the above methods is implicitly using some form of the weighted-mean method. We say this because giving some members of a committee, in effect, "veto" power represents in a very practical manner the assignment of importance not given other members.

MAN-MACHINE INTERACTIONS

In previous papers we presented results of simulated man-machine interactions (Robinson, 1984; Robinson et al, 1985a; Robinson, 1986a). The simulations illustrated the behavior of the process but did not provide an opportunity to investigate semantic variation among and between users of geographic information systems. Recently we presented results of a session with SRAS by a so-called expert (Robinson et al, 1986c) and showed that there was significant semantic variation within a single, expert user. In addition, it was shown explicitly that intransitivities exist in the definition of a simple concept by an expert.

Man-Machine Sessions

In this paper we will discuss premliminary results obtained from 5 subjects, one of which was the expert referred to above. There were a total of 16 sessions with each subject. Two geographic databases were used, each providing a distinctly different spatial context. One database contained 29 settlements drawn from the 1:250,000 USGS map sheet for Waycross, Georgia. The other database consisted of 112 settlement locations drawn from the 1:250,000 USGS map sheet for Hartford, Connecticut.

For each database there were a total of 8 sessions. Each session was concerned with acquiring a spatial relation expressed as one of terms in Table I. For example, in a session operating on the Hartford database the subject was asked, by SRAS -

Is Port_Ewen Close_to Waterbury ?

Each subject responded with "yes" or "no." It is important to realize that the subject did have the map sheet available for reference. Terms and databases were presented to the subject in a randomized order. Generally sessions were separated by 24 hours or more. Before starting the sessions the subject was not informed of the term set and the subject was not aware of what term was to be covered in the next session. The general problem remained the same, so the only variables were terms and database.

Results and Discussion

For each subject and session we can generate the question-answer tree such as one shown in Figure 1. Each of these trees tells us how many steps the session took, which settlements were used as question units, in what order they were asked, and the response of the subject. If a subject answered in exactly the same manner as another subject, the tree will be exactly the same for both subjects. Furthermore, for concepts that are compliments such as "near" and "far," exactly complimentary responses yield the same tree differing only in responses.

As a rough indication of the level of semantic variation we can look at variations in length of sessions without regard to similarity of pattern. Table 2 shows how often sessions of particular length occurred. What is surprising is the great range in session length. Of 80 sessions there were 17 different lengths of question-answer sessions. This implies a fair amount of variation in question-answer patterns.

Given the considerable difference in database size we might expect the variations in the question-answer trees to be greater with larger databases. Of the 40 sessions per database, there were 17 unique question-answer trees resulting from use of the Waycross, Georgia database and 18 unique trees from use of the Hartford, Connecticut database. These results suggest two things. One is that size of the database may not influence the overall semantic variation. The other is that regardless of database size, there is a roughly even chance that one user's question-answer tree will be the same as anothers. This variation becomes even more pronounced when we break the results down according to the terms in Table 1.

Tables 3 and 4 illustrate how often there is agreement by the subjects on the definition of a term. Even if there is agreement, there is never more than 3 of the 5 in agreement. Some interesting observations can be made regarding those terms on which there is no agreement. Using the Georgia database, each subject had different definitions for the terms Far, Distant_from, and Short_distance_from. There is no agreement on the definition of in_the_Vicinity_of, Remote_from, and Close_to when using the Connecticut database. It is apparent from these results that one can expect little agreement on the exact definition of simple spatial relations. This leads us naturally into the topic of multiperson concept construction.

Table 5 shows the results of sessions for five subjects regarding the specification of the term Close_to. This term was chosen as a subject of special attention because there is an intransitivity imbedded in the concept of Close_to for subject 1. Also, notice that subjects 2 and 3 agreed exactly on the specification of Close_to. Specification of Close_to by Subject 4 is the most liberal of the five subjects. Note the preponderance of membership values greater than 0.5 and absence of any membership values equal to 0.00. The resulting pattern of membership values for Subject 5 resemble the results for Subject 1, but contains some differences and lacks the aforementioned intransitivity.

Responses to the questions regarding Close_to Douglas lead to identification of an intransitivity. In the question/answer process of Subject 1, Nicholls received a no response and Pearson received a yes response. Pearson is farther from Douglas than is Nicholls, yet the subject said that Nicholls is not close while Pearson is close. Since the map sheet was available it is appropriate to consider whether transport routes and/or major landscape features may have influenced

Table 1. Term Set Used in Man-Machine Interaction Sessions

Nearness Terms	Farness Terms
Near	Far
in_the_Vicinity_of	Remote_from
Close_to	Distant_from
Short_distance_from	Long_distance_from

Table 2. Length of Sessions and Their Frequency of Occurrence.

Number of Steps	Number of Sessions
2	9
4	5
6	3
9	1
10	6
11	16
12	1
14	14
15	2
16	6
17	4
37	4
40	1
43	2
45	1
47	4
48	1

Source: Author's calculations.

Table 3. Frequency of Matching Question-Answer Trees by Linguistic Term for Sessions Using the Georgia Database.

Term	Number of Agreements	Number of Steps
Near	2	14
in_the_Vicinity_of	2	14
Remote_from	2	10
Close_to	2	14
Long_distance_from	3	14

Source: Author's calculations.

Table 4. Frequency of Matching Question-Answer Trees by Linguistic Term for Sessions Using the Connecticut Database.

Term	Number of Agreements	Number of Steps
Near	2	11
Far	2	11
Distant_from	3	11
Short_distance_from	2	47
Long_distance_from	3	11

Source: Author's calculations.

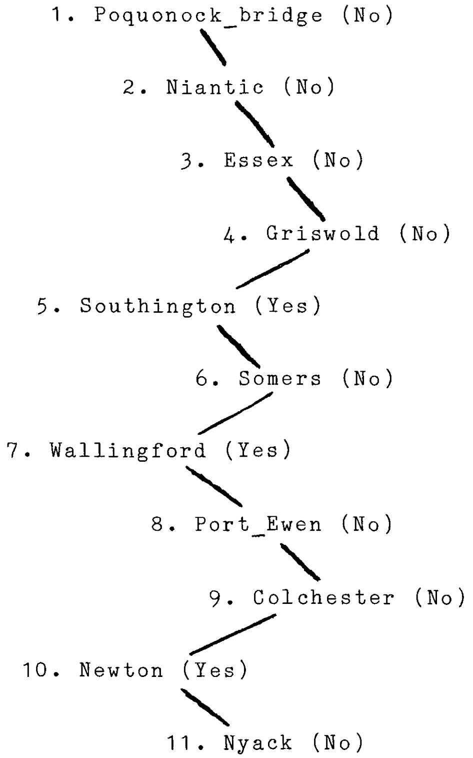


Figure 1. The Question-Answer Tree For A Subject's Definition of Near_Waterbury, Connecticut.

the subject. We find no major landscape features and both settlements are on direct, straight routes from Douglas. Furthermore, their routes are nearly orthogonal to one another. Depending on how one composes a multiperson concept, variations such as this intransitivity may be incorporated into multiperson concept.

In Table 6 we show what the results of several methods of multiperson concept formation using the membership values in Table 5 and using Subject 1 as the "chair of the committee." As one might expect, Subject 4 had an inordinate influence on the concept formed using the global evidence method. This was due to the preponderonce of high membership values. To dampen the influence of just such a situation the Combination Methods I and II were devised.

By inspecting the results in regard to the membership values of tifton, lenox, nicholls, and alma, one can see why method I is said to give the chairman 'veto' powers whereas method II is more like 'acceptance' powers. We chose Subject 1 as

Table 5. Fuzzy Membership Values for the Spatial Relation Close_to_Douglas Acquired from Subjects Using the Waycross, Georgia Database.

Settlement Name	Subjects 1	2	3	4	5
cordele	0.50	0.00	0.00	0.67	0.50
ashburn	0.53	0.40	0.40	0.70	0.50
sylvester	0.50	0.50	0.50	0.66	0.50
doerun	0.50	0.50	0.50	0.63	0.50
moultrie	0.52	0.52	0.52	0.64	0.50
coolidge	0.50	0.50	0.50	0.62	0.50
poulan	0.50	0.52	0.52	0.67	0.50
tifton	0.00	0.62	0.62	0.72	0.51
sycamore	0.48	0.44	0.44	0.70	0.50
lenox	0.43	0.65	0.65	0.71	0.51
adel	0.44	0.63	0.63	0.70	0.44
nashville	0.00	0.51	0.51	0.74	0.00
alapaha	0.00	0.00	0.00	0.76	0.00
ocilla	0.05	0.00	0.00	0.78	0.00
fitzgerald	0.00	0.00	0.00	0.78	0.00
rochelle	0.52	0.39	0.39	0.74	0.50
abeville	0.54	0.51	0.51	0.76	0.54
broxton	0.95	0.95	0.95	0.95	0.95
pearson	0.95	0.95	0.95	0.81	0.00
lakeland	0.00	0.00	0.00	0.73	0.00
willacoochee	0.52	0.52	0.52	0.79	0.00
homerville	0.60	0.60	0.59	0.77	0.00
nicholls	0.05	0.95	0.95	1.00	0.95
lumber_city	0.00	0.00	0.00	0.90	0.00
hazlehurst	0.00	0.00	0.00	0.93	0.00
waycross	0.05	0.05	0.05	0.05	0.05
blackshear	0.05	0.05	0.05	0.05	0.05
alma	0.00	0.72	0.72	0.92	0.56
baxley	0.00	0.05	0.05	1.00	0.00

Source: Author's calculations.

chairman to illustrate the importance of the chair in each of the methods. In Combination Method I the intransitivity regarding pearson and nicholls is perserved due to the veto power of the chair, whereas using method II eliminates the intransitivity. Also, note that in 24 cases the methods produced the same membership value, differing in those cases where substantial disagreement existed between the chairman and the remainder of the 'committee.'

We used the Zimmermann method with a number of different values of γ . Results the Zimmermann method are reported in Table 7. What is most disturbing about this method of multiperson concept formation is the influence that a membership value of zero has on it. Thus, we feel that this method is more appropriate when the membership values are

Table 6. Membership Values Resulting from Four Methods of Multiperson Concept Formation Using Results Reported in Table 5.

		Global		
Settlement	Agreement	Evidence	Combination*	
Name	Method	Method	Method I	Method II
cordele	0.00	0.67	0.50	0.50
ashburn	0.40	0.84	0.53	0.53
sylvester	0.50	0.79	0.50	0.50
doerun	0.50	0.74	0.50	0.50
moultrie	0.50	0.75	0.52	0.50
coolidge	0.50	0.70	0.50	0.50
poulan	0.50	0.80	0.50	0.50
tifton	0.00	0.89	0.00	0.51
sycamore	0.44	0.85	0.48	0.48
lenox	0.43	0.84	0.43	0.51
adel	0.44	0.80	0.44	0.44
nashville	0.00	0.84	0.00	0.00
alapaha	0.00	0.76	0.00	0.00
ocilla	0.00	1.00	0.05	0.05
fitzgerald	0.00	0.94	0.00	0.00
rochelle	0.39	0.83	0.52	0.52
abeville	0.51	0.83	0.54	0.54
broxton	0.95	0.95	0.95	0.95
pearson	0.00	0.95	0.95	0.95
lakeland	0.00	0.73	0.00	0.00
willacoochee	0.00	0.87	0.52	0.52
homerville	0.00	0.77	0.60	0.60
nicholls	0.05	1.00	0.05	0.95
lumber_city	0.00	0.90	0.00	0.00
hazlehurst	0.00	0.93	0.00	0.00
waycross	0.05	0.05	0.05	0.05
blackshear	0.05	0.05	0.05	0.05
alma	0.00	0.92	0.00	0.56
baxley	0.00	1.00	0.00	0.00

* Subject 1 was used as the "committee chairperson."
Source: Author's calculations.

non-zero. As noted above, one can plainly see that as increases so does the influence of the global evidence method. However, it remains unclear what guidelines should be followed when deciding on a value for γ.

Table 7. Results of Zimmermann's Method of Multiperson Concept Formation Using Membership Values in Table 5.

	Zimmermann Method				
Settlement			γ =		
Name	0.9	0.7	0.5	0.3	0.1
cordele	0.00	0.00	0.00	0.00	0.00
ashburn	0.43	0.50	0.58	0.68	0.78
sylvester	0.52	0.57	0.63	0.69	0.76
doerun	0.52	0.56	0.61	0.66	0.71
moultrie	0.52	0.56	0.61	0.66	0.72
coolidge	0.52	0.55	0.59	0.63	0.68
poulan	0.52	0.57	0.63	0.70	0.77
tifton	0.00	0.00	0.00	0.00	0.00
sycamore	0.47	0.53	0.62	0.70	0.80
lenox	0.46	0.52	0.59	0.68	0.78
adel	0.47	0.53	0.59	0.67	0.76
nashville	0.00	0.00	0.00	0.00	0.00
alapaha	0.00	0.00	0.00	0.00	0.00
ocilla	0.00	0.00	0.00	0.00	0.00
fitzgerald	0.00	0.00	0.00	0.00	0.00
rochelle	0.42	0.49	0.57	0.66	0.77
abeville	0.53	0.59	0.65	0.72	0.79
broxton	0.95	0.95	0.95	0.95	0.95
pearson	0.00	0.00	0.00	0.00	0.00
lakeland	0.00	0.00	0.00	0.00	0.00
willacoochee	0.00	0.00	0.00	0.00	0.00
homerville	0.00	0.00	0.00	0.00	0.00
nicholls	0.06	0.11	0.21	0.39	0.73
lumber_city	0.00	0.00	0.00	0.00	0.00
hazlehurst	0.00	0.00	0.00	0.00	0.00
waycross	0.05	0.05	0.05	0.05	0.05
blackshear	0.05	0.05	0.05	0.05	0.05
alma	0.00	0.00	0.00	0.00	0.00
baxley	0.00	0.00	0.00	0.00	0.00

Source: Author's calculations.

CONCLUDING COMMENTS

We demonstrated that approximate representations can be acquired from exact user responses using mixed-initiative man-machine interactions. In previous reports (Robinson et al, 1985a; Robinson et al, 1986a) we contented that human users might not respond in as deterministic manner as did the simulations. Results reported here and in Robinson et al (1986c) support that contention. In fact, our results suggest that even with a relatively small, simple spatial database, significant semantic variation does exist. This result has significant implications for maintaining semantic integrity within and between geographic databases. Furthermore, these results question the wisdom of using simple term matching to represent 'fuzzy' queries (eg. Chang and Ke, 1979).

These results show that there exists significant interuser and interterm variation. Results of this study suggest that size of the database may not influence the overall interuser semantic variation as much as one might suspect. In fact, this study suggests that regardless of database size, there is a roughly even chance that one user's question-answer tree will be the same as anothers. However, given our sample size, we refrain from making any strong probabilistic statements.

Of particular importance in this study has been the consideration of multiperson concept formation. Most spatial information systems are used in an organizational context where group (multiperson) concepts are the norm. We suspect that some of the work dealing with of multiperson concept formation will become valuable in dealing with semantic variations found in the distributed geographic information systems of the future. Regardless of the application domain, we showed how important it is to understand the process of multiperson concept formation. Future research on this topic will, most likely, be cast within the context of knowledge acquisition for use in expert systems.

Finally, we believe this study illustrates that this avenue of research has implications for developing systems for the detection and representation of ill-defined spatial entities as well as spatial relations that are by their nature fuzzy concepts. The mixed-initiative man-machine interaction coupled with approximate reasoning appears to us to be a particularly attractive approach for acquiring an approximate representation of ill-defined geographic concepts or features for subsequent use in an expert geographic information system.

ACKNOWLEDGEMENTS

Partial support was provided by National Science Foundation Grant IST-8412406 and a Research Grant from PSC-CUNY. We also wish to acknowledge the support provided by The University of Calgary and the City University of New York.

REFERENCES

Barrera, R. and A. Buchmann. 1981, "Schema Definition and Query Language for a Geographical Database System", IEEE Trans. on Computer Architecture, Pattern Analysis and Image Database Management, 250-256.

Buckles, B.P. and F.E. Petry. 1983, "Information-Theoretical Characterization of Fuzzy Relational Databases", IEEE Trans. Syst., Man, and Cybernetics, SMC-13, 74-77.

Chamberlain, D.D. and R.F. Boyce. 1974, "SEQUEL: A Structured English Query Language", in Proceedings of ACM SIGFIDET Workshop on Data Description, Access, and Control, Ann Arbor, MI., 713-775.

Chang, S. and J. Ke. 1979, "Translation of Fuzzy Queries for Relational Database System," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1, 281-294.

De Luca, A. and S. Termini. 1972, "A Definition of a Nonprobabilistic Entropy in the Setting of Fuzzy Sets Theory," *Information and Control*, 20, 301-312.

Frank, A. 1982, "MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation", *Computer Graphics*, 16, 199-207.

Gaglio, S., R. Minciardi, and P.P. Puliafito. 1985, "Multiperson Decision Aspects in the Construction of Expert Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, 15, 193-204.

Hersh, H.M., A. Caramazza, and H.H. Brownell. 1979, "Effects of context on fuzzy membership functions", in *Advances in Fuzzy Set Theory and Applications*, M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.), North Holland Publishing, New York, 389-408.

Jain, R. 1980, "Fuzzyism and real world problems," in *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, S.K. Chang and P.P. Wang (eds.), Plenum: New York, 129-133.

Kaufmann, A. 1975, *Introduction to the theory of fuzzy subsets*, Academic Press, New York.

Leung, Y. 1982, "Approximate characterization of some fundamental concepts of spatial analysis," *Geographical Analysis*, 14(1), 29-40.

Lundberg, G. 1982, "Modeling constraints and anticipation: linguistic variables, foresight-hindsight, and relative alternative attractiveness," *Geographical Analysis*, 14(4), 347-355.

Nakamura, K. and S. Iwai. 1982, "Topological Fuzzy Sets as a Quantitative Description of Analogical Inference and Its Application to Question-Answering Systems for Information Retrieval", *IEEE Trans. on Systems, Man, and Cybernetics*, 12, 193-204.

Robinson, V.B. 1983, "Integrating mathematical spatial algorithms in an interactive microcomputing environment," *Monograph* on Spatial Mathematical Algorithms for Land Data Systems,Lincoln Institute of Land Policy Colloquium Series: Cambridge, Massachusetts.

Robinson, V.B. 1984, "Modeling Inexactness in Spatial Information Systems", Proceedings of Pittsburgh Conference on Modeling and Simulation, Pittsburgh, PA.

Robinson, V.B. 1986, "Implications of Fuzzy Set Theory for Geographic Databases," Computers, Environment, and Urban Systems, in press.

Robinson, V.B. and A.U. Frank. 1985, "About Different Kinds of Uncertainty in Geographic Information Systems,", Proceedings, AUTOCARTO 7, Washington, D.C.

Robinson. V.B., D. Thongs, and M. Blaze. 1985a, "Machine Acquisition and Representation of Natural Language Concepts for Geographic Information Retrieval," Proceedings, Pittsburgh Modeling & Simulation Conference, University of Pittsburgh.

Robinson. V.B., M. Blaze, and D. Thongs. 1985b, "Natural Language Concepts in Geographic Data Processing Systems," Proceedings Intern'l Conf on Adv. Technology for Monitoring and Processing Global Environ'l Data, London, UK.

Robinson, V.B., M. Blaze, and D. Thongs. 1986a, "Man-Machine Interaction for Acquisition of Spatial Relations as Natural Language Concepts," in B. Opitz (ed.) Geographic Information Systems in Government, A.Deepak Press: Hampton, VA.

Robinson, V.B., A.U. Frank and M. Blaze. 1986b, "Expert Systems And Geographic Information Systems: Review and Prospects," Jrnl of Surveying Engineering, 112(2): 119-130.

Robinson, V.B., M. Blaze, and D. Thongs. 1986c, "Representation and Acquisition of a Natural Language Relation for Spatial Information Retrieval," Proceedings 2nd Intern'l Symp on Spatial Data Handling, 472-487.

Tomlin, C.D. and S.M. Tomlin. 1981, "An Overlay Mapping Language", presented at Symp on Regional Landscape Planning, Am. Soc. of Landscape Architects.

Yager, R.R. 1981, "Measurement of Properties on Fuzzy Sets and Possibility Distribution," Proceedings 3rd Intern'l Seminar on Fuzzy Set Theory, E.P. Klement (ed), J. Kepler Univ, Linz, Austria, 211-222.

Zadeh, L. 1978, "PRUF-A Meaning Representation Language for Natural Languages," Int. Jrnl. Man-Machine Studies, 10, 395-460.

Zadeh, L. 1981, "Test-Score Semantics for Natural Languages and Meaning Representation via PRUF," in Empirical Semantics, B. Rieger (ed.), Brockmeyer, Bochum, 281-349.

Zimmermann, H.J. 1983, "Using Fuzzy Sets in Operational Research," European Jrnl of Operational Research, 13(3): 201-216.

ACCESSING LARGE SPATIAL DATA BASES VIA MICROCOMPUTER

Christopher G. Heivly
Office of The Geographer
Department of State
Washington, D.C. 20520

Timothy White
Social & Behavioral Sciences Lab
University of South Carolina
Columbia, S.C. 29208

ABSTRACT

The Office of The Geographer within the Department of State is currently changing its production mapping unit from a manually intensive system to a computer intensive system. At the core of the computer mapping system is a number of microcomputer workstations. The workstations consist of IBM-AT's with high resolution graphic boards and monitors, digitizers and AUTOCAD software.

This paper discusses the utilization of the World Data Bank II data within this microcomputer framework. This data is normally used in large mainframe graphic systems. However, the data has been reformatted and reconfigures for fast, efficient interactive use on the micro-workstations.

INTRODUCTION

The Office of The Geographer, Department of State is responsible for producing production quality maps and graphs for various bureaus within the State Department. These maps and graphs are produced for every day publications and some long term publications as well. The turnaround time for producing maps and other graphics on a timely basis is a problem that has plagued the office for some time. To rectify the time spent drafting maps in a manual mode, the office has implemented a PC based system designed to create maps for any part of the world.

Data

The move towards automation of map and graphic production began in February of 1986. The first priority of the office was to build a system that could access digital data of the entire world, place the data within a graphic framework for additional manipulation and annotation, and to have a mechanism to output the final maps in various formats. Data requirements were simple:

1. Coverage of the entire world for coastlines and international boundaries,
2. Data in latitude and longitude coordinates.

The primary use of the digital data is to provide a base map to which other important information can be added during the annotation process. Based on these simple requirements the CIA's World Data Bank files were selected. These files represent the entire world in a latitude/longitude format and representation for four areas of the world (North America, South America/Antarctica, Europe/Africa, and Asia). World Data Bank I includes:

1. Coastlines, Islands and Lakes, and
2. International boundaries.

The average digitized map scale was 1:12,000,000. The files contain approximately 110,000 points representing the two overlays. World Data Bank II includes:

1. Coastlines, Islands and Lakes,
2. International Boundaries,
3. Internal Administrative Boundaries,
4. Rivers,
5. Roads, and
6. Railroads.

The average digitized map was 1:3,000,000. The files contain approximately 10,000,000 points representing the six overlays (Coverage is not complete for features such as roads, railroads, and internal boundaries).

Hardware

The hardware for the map production system consists of two workstations. These configurations are:

WORKSTATION #1

1 - IBM-AT; 640K, (2) 20 meg hard disks, 1.2 meg floppy disk, 80287 math co-precessor.
1 - ARTIST 1+ graphics card; 1024x1024 resolution, 16 colors.
1 - GIGATEK 19" color monitor.
1 - CALCOMP 1043GT plotter; A-E size plots, 8 pen.
1 - GTCO digitizer; 36X48", 16 button.
1 - OKIDATA Microline 193 printer.
1 - ALLOY 9-Track tape drive.
1 - Bernoulli Box; (2) 20 meg removable cartridges.

WORKSTATION #2

1 - IBM-AT; 640K, 20 meg hard disk, 1.2 meg floppy disk, 80287 math co-processor.
1 - TECMAR graphics card; 640x350 resolution, 16 colors.
1 - BITEC 13" color monitor.
1 - Hewlett Packard plotter; A-B size, 6 pen.

Software

The software utilized on the production mapping workstation is AUTOCAD from AutoDesk, Inc. AUTOCAD provides a vector based graphics toolbox containing various functions to manipulate graphic features including a wide range of annotation capabilities. AUTOCAD also contains device drivers for a multitude of graphics boards, monitors, digitizers, plotters and printers.

MAP PRODUCTION SYSTEM

The map production system as designed contains four basic sections; (1)Data conversion, (2) Data query software including download capability from the 9-Track tape drive, (3) Projection and AUTOCAD conversion, and (4) interactive computer aided map design.

Data Conversion

World Data Bank files represent features of the world in a simple vector format. The direct access format for mainframe system using CAM/GS-CAM contains two types of files. The first is an index file containing line id number, feature rank code, number of points for the line feature, coordinate data file. Each index record is binary, unformatted and 32 bytes long. The second file type is a coordinate file containing 50 pairs of llatitude/longitude points in radians. Direct access records are binary, unformatted and 400 bytes long (50 coordinates per record * 2 words per coordinate * 4 bytes per word).

This original format is a simple but inefficient storage format which performs adequately for mainframe graphics systems. Access speed, and storage are not serious issues for a mainframe perspective. However, in a microcomputer environment this data structure does not optimally utilize existing disk space. Specifically, when the number of points in a linear feature is not an even multiple of 50, space will be wasted in the file. For example, given two features, the first with 163 points, the second containing 52 points, the following coordinate assignment would occur;

RECORD 1	---	coordinates	1- 50	for feature #1.
RECORD 2	---	coordinates	51-100	for feature #1.
RECORD 3	---	coordinates	101-150	for feature #1.
RECORD 4	---	coordinates	151-163	for feature #1.
RECORD 5	---	coordinates	1- 50	for feature #2.
RECORD 6	---	coordinates	51- 52	for feature #2.

In this example, only 26% of record 4 and 4% of record 6 is used. In other words, a total of 680 bytes of potential coordinate space is wasted.

In a microcomputer environment, the limitations of internal memory (RAM) and external storage (Disk space) made the processing of the original WDBII format very time consuming. Thus, an alternative data format was needed to alleviate the storage and access speed problems.

There were two basic operations performed on the WDBII files to enhance access speed and disk storage. The first operation attempted to compress the overall size of the files without eliminating any points. A revised direct access format was created by processing the sequential data structure one feature at a time. For each line output is generated for two data files; an index file, and a coordinate file. This is similar to the original CAM/GS-CAM direct access format. The new index record contains alternative information about each line feature. This information includes feature rank code, number of points representing the line, an index value indicating starting record position, an offset indicating the relative position within the record, and two coordinate pairs defining the feature window. Eliminated from the original format were the line id number. Each index record is binary and 32 bytes long. Additional disk access speed was gained by changing the block size of the coordinate and index files. Disk space on the IBM PC/XT/AT is allocated in clusters. A cluster is a group of disk sectors which varies in size from one to four kilobytes depending on the recording media (i.e. floppy disk, XT hard disk, or AT hard disk) and version of Disk Operating System (DOS). By formatting the WDBII data into records which correspond to cluster size results in the fastest and most efficient disk access possible on the AT. For the AT hard disk and DOS version 3.1, the cluster size is 2048 bytes. A new record size of 2048 bytes is created with a blocking factor of 256 coordinate pairs per record (1 coordinate pair/8 bytes per pair * 2048 bytes) for the coordinate file. Index records are grouped together in units of 64 to form direct access records. By changing the structure of both files to access cluster size records, the total number of records has been reduced and dramatic increases in access speed have been gained. For example, given a file with three features, the first with 514 points, the second with 102 points, and the third with 263 points, the following coordinate assignments would be made;

RECORD 1	---	coordinates	1-256	for feature #1.
RECORD 2	---	coordinates	257-512	for feature #1.
RECORD 3	---	coordinates	513-514	for feature #1 and
		coordinates	1-102	for feature #2 and
		coordinates	1-152	for feature #3.
RECORD 4	---	coordinates	153-263	for feature #3.

For the Office of The Geographer, this revised format is the most efficient and compact format for WDBII data files.

Quick Index Files

A second major need was to further decrease the time required to extract specific features for different geographic areas. The solution to the problem was the development of "quick" index files based on geographic area and feature rank code. The geographic quick index file consists of a 2 byte variable (16 bits) representing 16 areas of the world. Each area consists of a geographic window defined by 45 degrees of longitude and 90 degrees of latitude. The 16 areas correspond to the 16 bits in the 2 byte variable. The configurations are as follows:

AREA #1	180W/N	-	135W/N	-----	BIT 1
AREA #2	134W/N	-	90W/N	-----	BIT 2
AREA #3	89W/N	-	45W/N	-----	BIT 3
AREA #4	44W/N	-	0W/N	-----	BIT 4
AREA #5	1E/N	-	45E/N	-----	BIT 5
AREA #6	46E/N	-	90E/N	-----	BIT 6
AREA #7	91E/N	-	135E/N	-----	BIT 7
AREA #8	136E/N	-	180E/N	-----	BIT 8
AREA #9	180W/S	-	135W/S	-----	BIT 9
AREA #10	134W/S	-	90W/S	-----	BIT 10
AREA #11	89W/S	-	45W/S	-----	BIT 11
AREA #12	44W/S	-	0W/S	-----	BIT 12
AREA #13	1E/S	-	45E/S	-----	BIT 13
AREA #14	46E/S	-	90E/S	-----	BIT 14
AREA #15	91E/S	-	135E/S	-----	BIT 15
AREA #16	136E/S	-	180E/S	-----	BIT 16.

As each line was processed, corresponding bits were turned on based on the extremes of the latitude and longitude coordinates. This 2 byte variable can be quickly scanned and a determination made whether the line falls within the geographic window selected by the cartographer. The quick index file was also formatted to meet the cluster size of the AT. Thus, 1024 two byte variables reside on a record.

A similar version was made for the feature rank code. This code corresponds to a detailed ranking for each overlay. For example, in the railroad file there are subcategories for broad gauge, standard gauge, and narrow gauge railroads. The rank index file contains a one byte variable that corresponds to the individual feature rank codes. This variable can be scanned to determine whether the code matches the code(s) selected by the cartographer. Both the geographic and the feature rank index files were created to speed the data access process. By eliminating lines not falling within the predescribed window, the amount of data to be transferred to the PC is greatly reduced.

Data Query System

The data query system is the data base accessing module. There are three basic data queries that the cartographer can perform; (1) geographic window, (2) scale data base, and (3) feature/rank extraction.

The "geographic window" parameters contain the lat/long coordinates of the window selected by the cartographer. This allows the query system to eliminate data from the data base that is not needed for the specific application. The "scale data base" refers to either WDBI or WDBII. Depending on the application/scale of the final map product, one of the two data bases is chosen. For example, a map of Africa generated at 8.5"x11" does not require the detail of data within WDBII. Likewise, a map of El Salvador at 30"x40" does require the greatest detail possible from WDBII.

Projection & Conversion

After the appropriate data has been extracted they must be projected into a cartesian coordinate system and converted into the AUTOCAD data interchange format. In order to accomplish this the USGS General Cartographic Transformation Package (GCTP) was transferred into the PC-AT environment and linked to the data query subsystem. GCTP provides excellent forward and reverse projection transformations between 20 different map projections. The resultant XY coordinate strings are then written out as AUTOCAD polylines. These polylines are easily imported into AUTOCAD. AUTOCAD provides a full range of computer aided design functions that facilitate the final map production process.

CONCLUSION

The reformatting and reconfiguration of WDBII data files has provided the Office of The Geographer with a faster, more efficient data format from which microcomputer workstations can access "mainframe" type data. The ability for fast search and retrieval methods to access this data has greatly reduced the turnaround time for producing production quality maps. The fast access of digital data combined with the ability to transform the data into 20 different projections and work in an interactive CAD environment enables the cartographer to quickly and more accurately display geographic information.

Future developments include the polygonization of the WDBII files which involves adding topology to the data structure, accessing other world-based digital files within a similar framework, and an interactive projection driver to quickly view the results of the projection parameters to verify the geographic window and projection nuances.

REFERENCES

Software Documentation for GCTP: General Cartographic Transformation Package, U.S. Geological Survey, National Mapping Division.

GS-CAM: Geological Survey-Cartographic Automatic Mapping, U.S. Geological Survey, National Mapping Division.

CAM: Cartographic Automatic Mapping Program Documentation, Central Intelligence Agency.

AUTOCAD User Reference Manual.

A GENERAL APPROACH TO MAP CONFLATION

Anthony E. Lupien
William H. Moreland

Environmental Systems Research Institute
Redlands, California 92373

Abstract

Map conflation is approached as two generic geoprocessing problems, feature alignment and feature matching. Techniques for their solution are presented. By taking a general approach, match relationships between maps can be exploited by relational database functions of a geographic information system to create various products. Examples are presented from the Arc/Info implementation.

Introduction

Lynch and Saalfeld define conflation as a "combining of two digital maps to produce a third map file which is 'better' than each of the component source maps (1985)." In their pioneering conflation experiments at the Census Bureau they developed a method of conflating maps from two particular series, GBF/DIME and USGS scanned maps. Their reports of this development provide a vocabulary for conflation research and numerous suggestions for implementing conflation in geographic information systems (GIS) software.

Many GIS applications can be served by a conflation capability. These include transfering attributes from old to new street network maps, insetting maps into existing spatial databases, unifying maps of different geographic features and evaluating maps' coordinate distortion. The edge-matching problem can even be treated as a subset of conflation.

This paper presents the method we have developed for conflation, which is treated as the solution of two generic geoprocessing problems within a relational database environment. Solution of the feature alignment and feature matching problems, it is shown, creates database relationships between two maps that may be used to generate any desired "third map."

In conflation, the "best" elements of two maps are combined to form a third. Definitions of what is "best" in a map vary with application, but whatever conflation product is sought conflation is generally the same problem: Identifying features in two maps that represent the same earth feature or are in the same location, then selectively merging features and attributes of both into a third.

Conflation as Two Generic Problems

Conflation requires solution of two generic geoprocessing problems: Feature alignment and feature matching. By "features" we mean the lines, points and polygons that comprise topological digital maps, or "coverages".

Features are aligned by transforming the coordinates of those in one coverage to fit another's. Rubber-sheeting techniques are used to solve this problem. Our implementation of rubber-sheeting involves simple extensions to an existing graphic editing program and a coordinate transformation program developed for modeling triangulated irregular networks (TIN). A data structure called a "link", which can be treated as both an arc and a point, is used to bridge the requirements of the two programs.

Once features of two coverages have been aligned they may be matched. Rosen and Saalfeld report success with iterative applications of topological criteria that compare features based on their relationships to others in the same coverage (Rosen and Saalfeld, 1985). Our research, however, suggests that simple feature-to-feature distance measurements between coverages suffice. In the Arc/Info implementation most map features that should ideally match generally do, without significant false-matching.

Solving the Feature Alignment Problem

We would like to believe our success in conflation with simple feature-matching rules is due to the strength of our feature alignment solution. The first component of this is an interactive graphic system for displaying two coverages and adding "links" to one of them. Each link can be thought of as an arc from a location in the coverage to be transformed to a corresponding location in the other (fig 1). As arcs, these links can be added, deleted, modified and symbolized using common GIS software. Link management capabilities were, therefore, easy to add to the graphic editing subsystem of Arc/Info.

These links are passed to a coverage coordinate transformer, without disturbing the graphics environment. This program treats links as points with delta-x and delta-y attributes and builds from them a triangulated irregular network. From this network two fifth-order distortion surfaces are constructed, one for x and one for y. The distortion surfaces are used to transform coverage features, which are redisplayed (fig 2). A non-linear (bivariate quintic) interpolation algorithm is used to produce smooth surfaces (Chen and Guevara, 1986).

Links are added and transformations run iteratively. Alignment of 500 to 1000 features, a useful working set size (Lynch and Saalfeld, 1985), is controlled with only a few links. Experiments (Lupien, 1986) suggest that one well-placed link per 50 features yields close alignment (fig 3).

After initial alignment further global transformations tend to cause aligned features to drift apart. To avoid this drifting phenomenon subsequent transformations are limited by use of zero-length, "identity", links. An identity link effectively "nails down" a coverage location, preventing it from moving in transformation.

Surrounding an area with identity links localizes the impact of links within the area (fig 4). Features inside the boundary will move; those outside will not. Identity links may also be used to limit transformation to a subset of coverage features selected by some attribute. After initial alignment and matching, transformations are often limited to unmatched features by placing identity links along matched features (fig 5).

The software for building TINs and using them to generate surfaces was, like the graphics editor, developed for Arc/Info without anticipating its usefulness in conflation. But by structuring the software as generic modules, and by recognizing links to be a conceptual bridge between graphical arcs and points of distortion, the actual development of strong feature alignment tools was made quite economical.

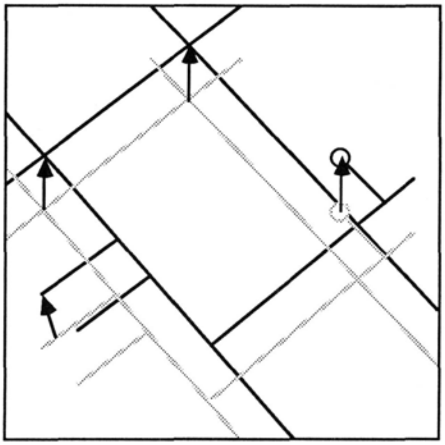


Figure 1

Each link can be thought of as an arc from a location in the coverage to be transformed to a corresponding location in the other.

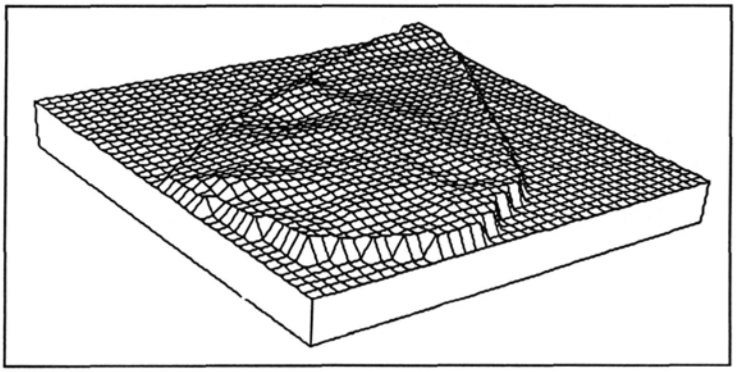


Figure 2a

50 link distortion surface for X

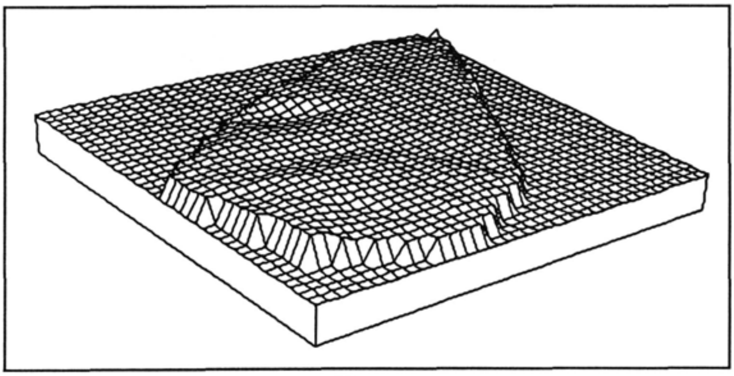


Figure 2b

50 link distortion surface for Y

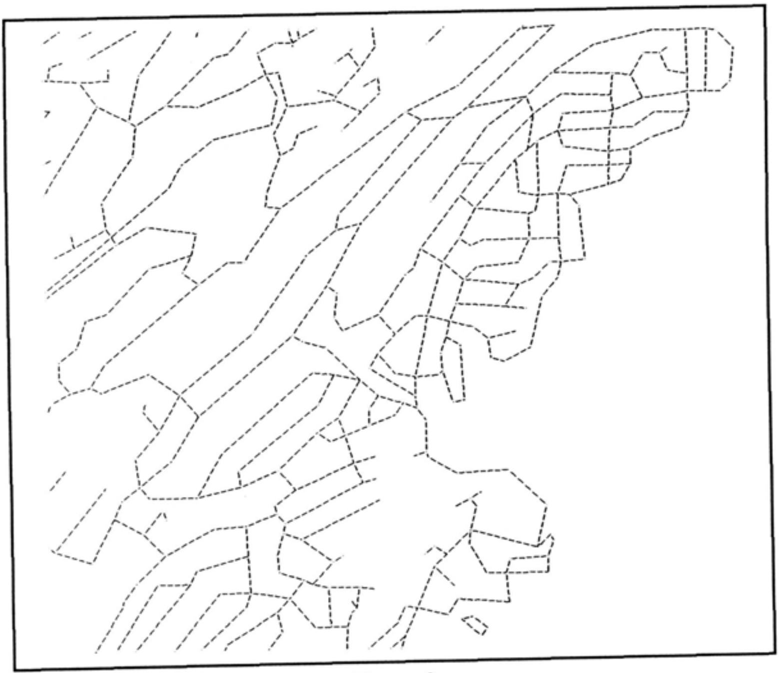


Figure 3a
Partial street network coverage, Atlanta GBF/DIME file.

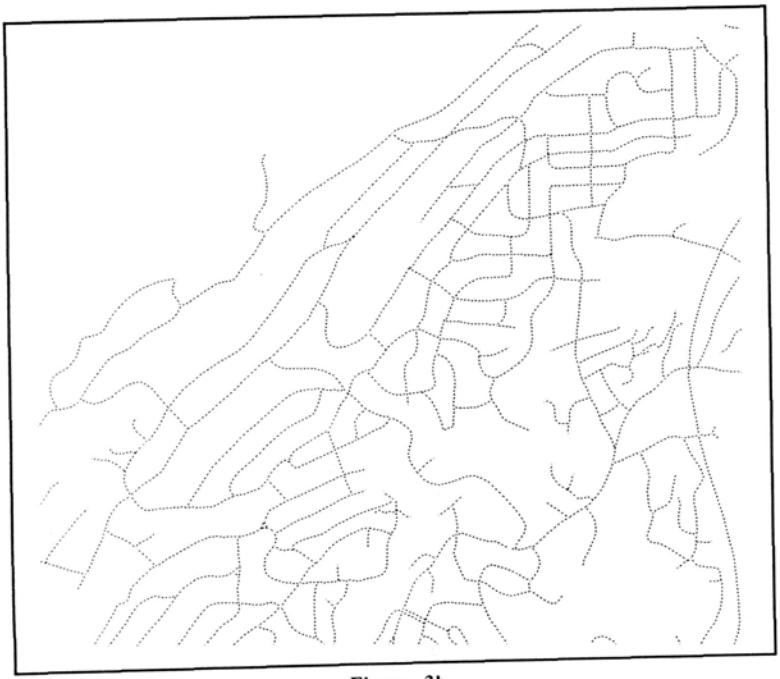


Figure 3b
Partial street network coverage, Courtesy BellSouth Services Co.

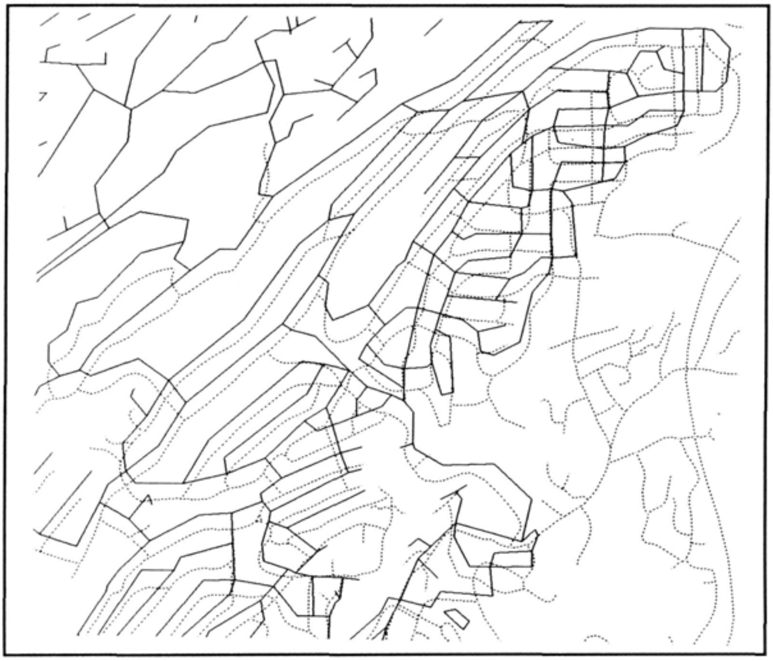


Figure 3c
Original coverage alignment (untransformed).

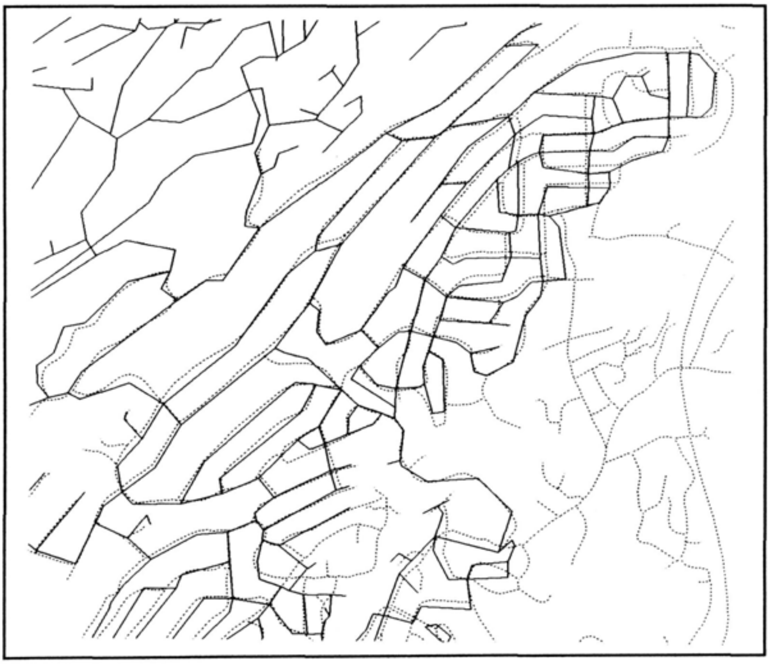


Figure 3d
Coverages aligned with 5 links.

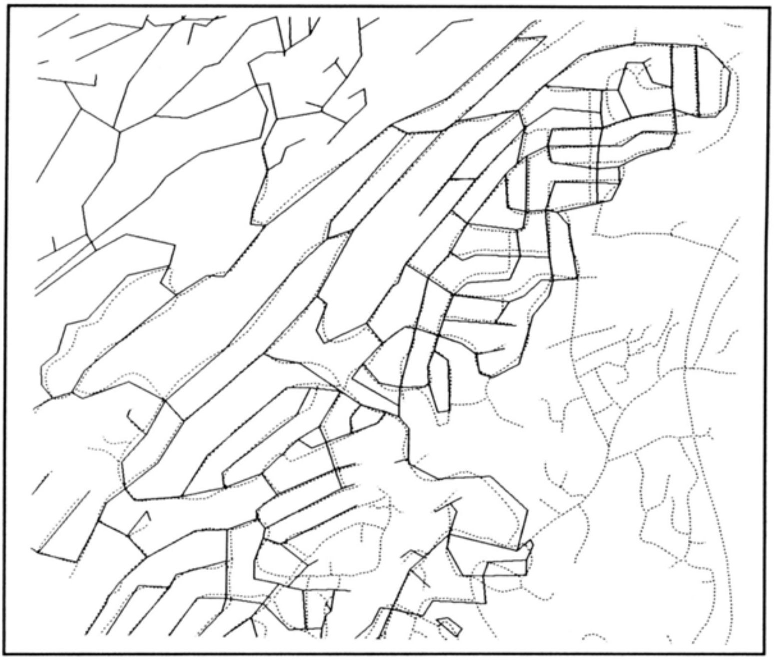


Figure 3e
Coverages aligned with 50 links.

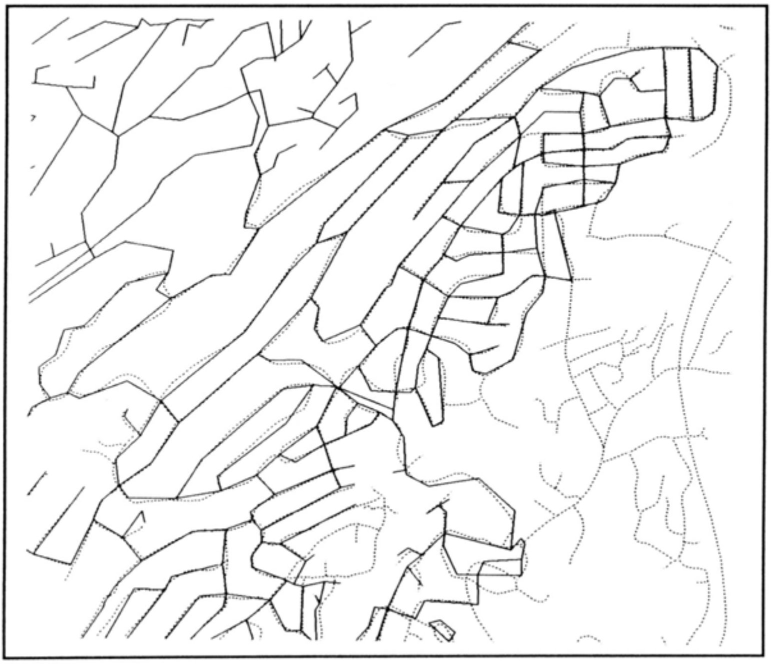


Figure 3f
Coverages aligned with 50 links, then locally adjusted (fully processed).

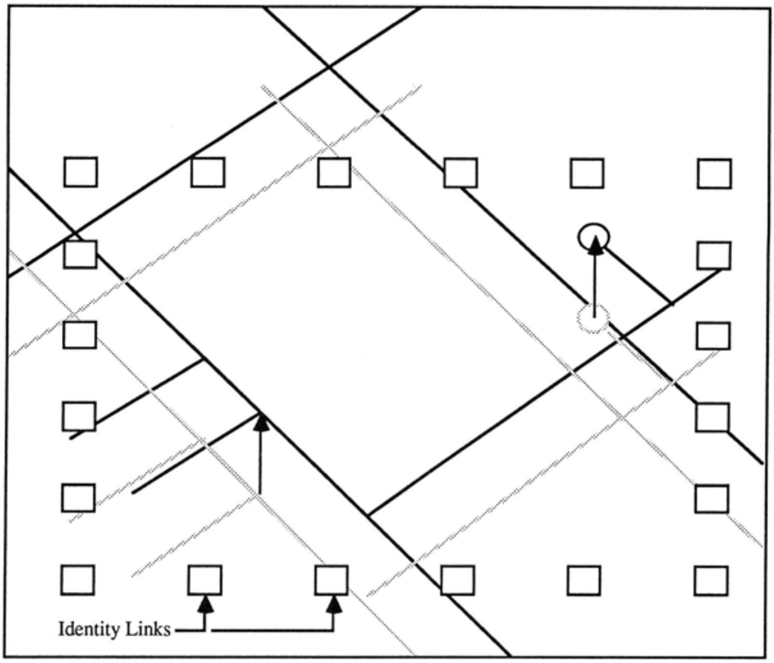


Figure 4

Surrounding an area with identity links localizes the impact of links within the area.

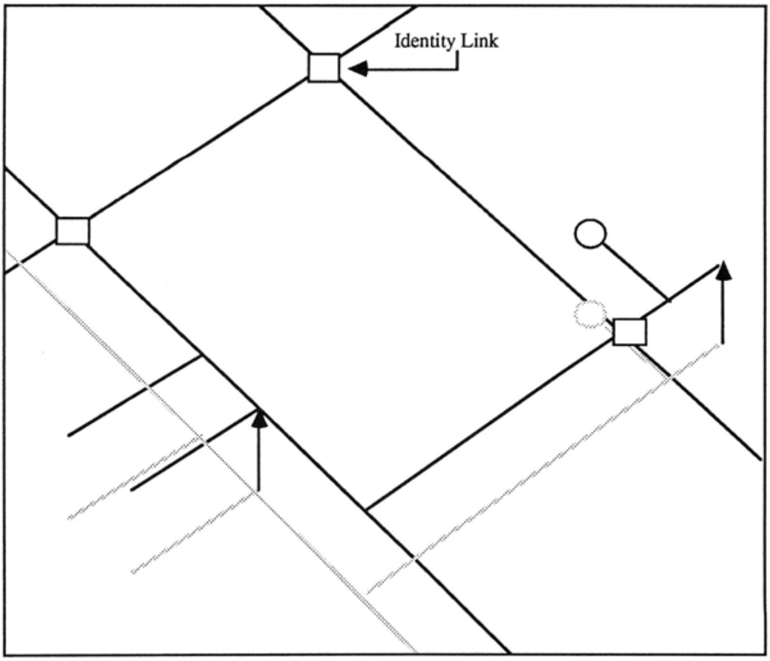


Figure 5

Transformations are limited by placing identity links along matched features.

Solving the Feature Matching Problem

Simple geometric distance calculations may be used to identify matching features in two closely-aligned coverages (fig 6). These calculations are computationally effecient, which is particularly desirable because of conflation's iterative nature. The criteria presented below are used to match point and arc features. In Arc/Info, polygon feature locations are represented by points and are best matched using point-in-polygon testing.

A point feature in coverage A is matched if only one point in coverage B is within a specified tolerance distance. This tolerance is relaxed in successive runs that follow alignment iterations, as suggested by Rosen and Saalfeld (1985).

Arc features are matched in the same way, except that distance calculations are made from each arc vertex, including its from- and to-nodes, to the nearest point along candidate arcs. If each vertex of an arc in coverage A is within tolerance of an arc in coverage B, and if this is true for only one arc in B, then a match is recorded.

These match criteria produce many-to-one relationships, which are sometimes desirable. To produce one-to-one matches only, the procedure is run in both directions (coverage A to coverage B, then coverage B to coverage A), and duplicate pairs extracted from the two lists of matches.

Products of Conflation

Once match relationships have been determined they may be used to transfer attributes between coverages, or to merge the unique features of conflated coverages into a third, or to create maps in which features of two coverages are selectively symbolized.

Many GIS applications, including the Census Bureau/USGS joint project (Lynch and Saalfeld, 1985), seek to transfer attributes from one coverage to another. Often this is done to improve the cartographic quality of an otherwise useful database.

Attribute transfer in the Arc/Info implementation is performed by relating the feature attribute tables of matched coverages to the normalized list ennumerating feature matches. The attribute tables of matching features are then joined.

Other applications seek to merge unique features of two coverages into a third. Many such efforts are aimed at updating a database with new information. Conflation is used to identify new features that have no equivalent in the existing database.

Conflated coverages are merged in Arc/Info by relating their feature attribute tables to the list of matches, then selecting those features in each coverage that have no matching feature in the other. The two selected sets are saved to a third coverage.

Conflation may be used in still other GIS applications to improve the quality of maps produced from several coverages. A map might be made, for example, using features from hydrographic and street-network coverages. Such a map might be produced at a scale that would place a river and a road in the same location. This can be avoided by conflating the coverages and using match relationships to control map symbolization.

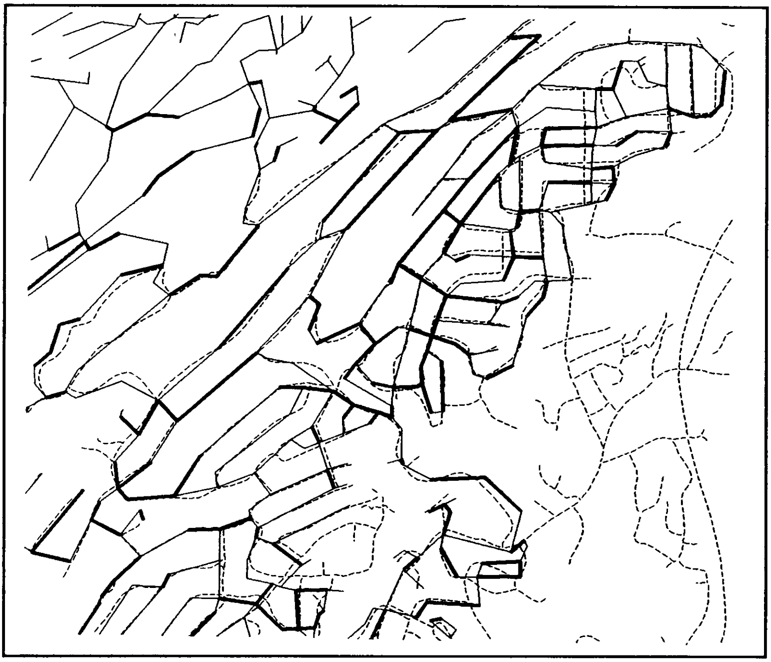


Figure 6a
Matching features after initial alignment.

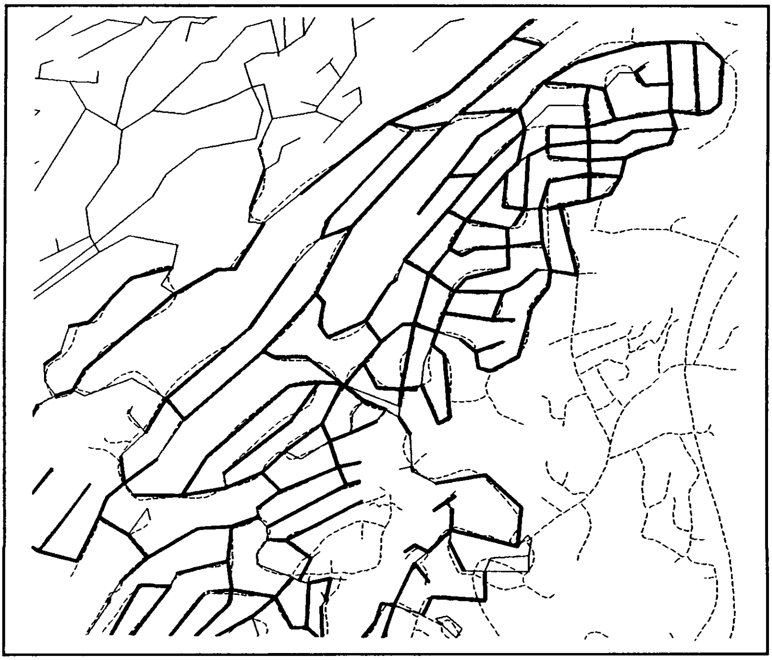


Figure 6b
Matching features after full processing.

References

Chen, Z. and A. Guevara, 1986, "System Selection of Very Important Points from Digital Terrain Model for Constructing Triangular Irregular Networks," *Proceedings, Auto-Carto VIII.*

Lupien, A., 1986, "Conflation with Arc/Info", internal Environmental Systems Research Institute document.

Lynch, M. and A. Saalfeld, 1985, "Conflation: Automated Map Compilation, a Video Game Approach", *Proceedings, Auto-Carto VII.*

Rosen, B. and A. Saalfeld, 1985, "Match Criteria for Automatic Alignment," *Proceedings, Auto-Carto VII.*

RASTER AND VECTOR PROCESSING FOR SCANNED LINEWORK*

David D. Greenlee
U.S. Geological Survey
EROS Data Center
Sioux Falls, South Dakota
57198

ABSTRACT

Recent advances in scanning technology have made it possible for linework to be scanned on relatively inexpensive systems and output as raster images. These images can be edited, automatically thinned to connected lines, and converted to vector representations using conventional image processing techniques.

An investigation of raster editing techniques, including thinning, skeletonizing, filling, and node detecting was performed by using software tests implemented on a microprocessor. The technique was based on encoding a three-by-three neighborhood surrounding each pixel into a single byte. This byte has a range of 0-255, and stores any possible surround condition for a pixel. It was found that various methods for thinning, filling, and node detection could be quickly implemented and tested using surrounding topology. It was possible to efficiently develop decision tables for raster editing, and to assure symmetry and consistency in the decisions.

A prototypical method was also developed for converting the edited raster linework into vectors. Once vector representations of the lines were formed, they could be formatted as a Digital Line Graph, and further refined by deletion of nonessential vertices and by smoothing with a curve-fitting technique.

INTRODUCTION

The conversion of rasterized linework to vector chains or arcs has provided researchers with a challenging set of problems for several years. Peuquet (1981) gives a technical overview of these methods. Based on a large body of available techniques, this paper describes the use and refinement of several methods for processing linework in raster and vector form. Automated aspects of the raster-to-vector process are described, but interactive editing steps are beyond the scope of this study.

This paper also attempts to bridge a gap between theoretical work on raster-to-vector processing and the many turn-key production systems that are in use but are often not well understood. During this project, techniques were developed that created gaps, spikes, spurs, snow, node slippage, and various other unsavory artifacts of automated processing. Through this experience has come a better understanding of the automated techniques that work best for a given application and some insight on the correct mix of automated versus manual editing techniques.

BACKGROUND

In support of applications project work, techniques have recently been developed by Jenson (1985) for extracting hydrologic basins and for delineating drainage networks from raster-formatted Digital Elevation Models (DEM's). It was

* Publication authorized by Director, U.S. Geological Survey

determined that a conversion of drainage lines to vector format might better allow for efficient overlay with other mapped data, for measuring lengths and areas, and for ordering of streams. Techniques were available for converting classed hydrologic basins into polygonal form (Nichols, 1983), but not for converting rasterized drainage lines into vector form.

A method was needed to convert rasterized linework, as extracted from the DEM, into vector chains or arcs that could be input to a vector-based geographic information system (GIS). It was observed that this problem is analogous to the problem of converting scanned map linework into vectors. By developing a generic solution to this problem, it was hoped that future scanning of map data could be facilitated by an understanding of these processing and conversion steps.

ENCODING OF TOPOLOGICAL SURROUNDS

Rasterized linework can be visualized as a bilevel or binary image having pixels represented by ones if a line is present, and represented by zeros if they are in the space between lines. In this form, many techniques are available for processing and refining the lines (Rosenfeld and Kak, 1982; Pavlidis, 1982). Most techniques utilize a set of rules for operating on each pixel based on its state and that of its eight contiguous neighbors. Because neighborhood operations are time consuming and relatively troublesome to perform, this was a subject that needed special attention.

The method described below was inspired by Golay (1969), who described 64 surround conditions for a hexagonal data structure. Since image data are usually in a regular raster format, the technique used here encodes for each pixel a single byte of eight bits that completely describe the state of adjacent neighbors. For raster-processing operations to be performed (for example, filling, thinning, or node detection), a set of decision rules is first developed and placed in a table with one entry for each possible surround condition. The raster image is then processed in the following order:

```
a) encode the surround condition for each pixel, and store it

b) for each pixel,
        1. find the decision rule referenced by the surround condition
        2. make changes to the pixel where indicated by the rule
        3. if changed, update the surround values for neighbor pixels

c) if any changes were made on this pass, go to b)
```

By encoding the surround values on the first pass and then updating them only when changes have been made, it is not necessary to encode the value on subsequent passes. This can greatly improve the speed of processing, since many of the processing functions are applied iteratively, until no changes are made in an entire pass.

Encoding can be performed by adding together the values shown on the following diagram for the neighbor pixels that have a line present.

64	128	1
32	X	2
16	8	4

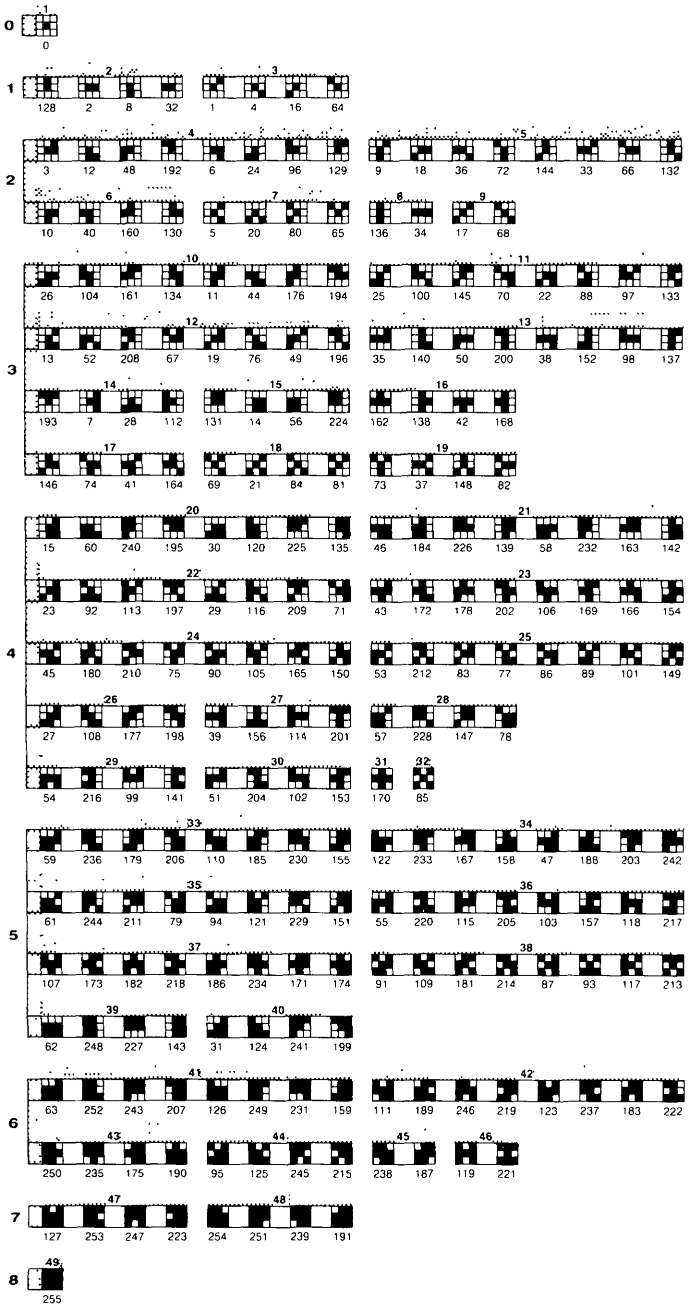


Figure 1 - The 256 possible *surround conditions* for a pixel and its eight adjacent neighbors

PAT GRP NUM	NUMBER OF 8 NGHBRS	PATS IN GRP	NUMBER OF TRANS	NUMBER OF GAPS	FILL RULE	THIN RULE STEP1	THIN LANDY STEP1	THIN RULE STEP2	THIN LANDY STEP2	NODE MARK RULE	NODE MARK SAKAI
1	0	1	0	0				Y		T	
2	1	4	1	1				Y		T	T
3	1	4	1	1				Y		T	T
4	2	8	1	1				Y		T	T
5	2	8	2	2							
6	2	4	2	1				Y	Y		
7	2	4	2	2							
8	2	2	2	2							
9	2	2	2	2							
10	3	8	2	1				Y	Y		C
11	3	8	2	2							C
12	3	8	2	2							C
13	3	8	2	2							C
14	3	4	1	1		Y	Y	Y	Y	T	
15	3	4	1	1		Y	Y	Y	Y		
16	3	4	3	1	Y			Y		C	M
17	3	4	3	2						C	M
18	3	4	3	3						C	M
19	3	4	3	3						C	M
20	4	8	1	1		Y	Y	Y	Y		
21	4	8	2	1	Y		Y	Y			C
22	4	8	2	2							C
23	4	8	3	1	Y			Y	Y	C	M
24	4	8	3	2						C	M
25	4	8	3	3						C	M
26	4	4	2	1				Y	Y		C
27	4	4	2	2							C
28	4	4	2	2						C	C
29	4	4	2	2							C
30	4	4	2	2							C
31	4	1	4	1	Y				Y	C	C
32	4	1	4	4						C	C
33	5	8	2	1	Y			Y	Y		C
34	5	8	2	1	Y			Y	Y		C
35	5	8	2	2						C	C
36	5	8	2	2							C
37	5	8	3	1	Y			Y	Y	C	M
38	5	8	3	3						C	M
39	5	4	1	1	Y	Y	Y	Y	Y		
40	5	4	1	1		Y	Y	Y	Y		
41	6	8	1	1	Y			Y	Y		
42	6	8	2	1	Y			Y	Y	C	C
43	6	4	2	1	Y				Y	C	C
44	6	4	2	2						C	C
45	6	2	2	1	Y				Y	C	C
46	6	2	2	2						C	C
47	7	4	1	1	Y						
48	7	4	1	1	Y						
49	8	1	1	1	Y						

Figure 2 - Pattern groupings of topologically similar surround conditions and their corresponding decision rules

Digitally, encoding can be performed by setting the corresponding binary bits of an eight-bit byte. In either case, the resulting value occupies the range of values 0-255, or 256 possible states. An example of a Fortran subroutine to perform this encoding would look like the following:

```
          SUBROUTINE ENCODN (NBR8,IVAL)
C
C     IVAL = output value (byte)
C     NBR8 = input array of eight    MASK = bit masks corresponding
C            neighbors ordered as:          to NBR8 as follows:
C              1  2  3                        64  128  1
C              4     5                        32       2
C              6  7  8                        16   8   4
C
           DIMENSION NBR8(8),MASK(8)
           LOGICAL*1 IVAL
           DATA MASK /64,128,1,32,2,16,8,4/
C
           IVAL = Ø
           DO 1ØØ K=1,8
                IF (NBR8(K).EQ.Ø) GO TO 1ØØ
                IVAL = IVAL.OR.MASK(K)
100        CONTINUE
           RETURN
           END
```

The complete set of 256 possible surround conditions is shown in Figure 1. In this figure, surround conditions have been grouped to allow decisions to be developed easily and consistantly. To accomplish this, the 256 surround conditions were combined into pattern groups by sorting on a set of attributes. Within each of the count groups (labelled 1-8 and organized vertically), are subgroups that may contain as many as eight patterns. These are referred to as pattern groups and may consist of a pattern, rotations of 90°, 180°, and 270°, and its mirror image and rotations. Because many of the patterns are symmetrical, they generate non-unique or duplicate patterns when rotated. As a result, some subgroups will contain only four, two, or as few as one unique pattern. Grouping allows the 256 possible surround conditions to be aggregated into a more manageable 49 pattern groups. None of the operations described in this paper requires subdivision below the pattern group level.

Each of the pattern groups can be represented by a set of attributes that allows decision tables to be efficiently developed. Figure 2 is a table of pattern groups and associated attributes. One basic attribute is the number of neighbors that have a line present. Also, for thinning operations to be performed, it is useful to know whether the pattern is classed as simple, (Rosenfeld and Kak, 1982) or non-multiple (Pavlidis, 1980). For this condition to be true, a pixel must have exactly one pixel or a connected set of pixels adjacent to it. Landy and Cohen (1985) use similar information, that they refer to as the number of black to white transitions present. In addition, they determine the number of lines that result when the center pixel is deleted. This is referred to as the number of gaps, because it measures whether a gap in connectivity would be created by thinning. These characteristics are constant for each pattern group, and were used to define and order the groups shown in figure 1.

RASTER PROCESSING

The following section describes raster operations that were performed as a part of this study, as applied to two sample data sets. These examples demonstrate the various raster and vector operations that were performed in the process of converting raster-linework to vector arcs.

Figure 3a is a Digital Elevation Model expressed as a shaded-relief display. Figure 3b shows rasterized linework that has been extracted from the DEM as described by Jenson (1985). The linework is portrayed as a binary image with lines as ones and background as zeros.

Figure 4a is a small portion of a topographic map, scanned using commercially available raster scanning equipment. The image is expressed in 256 possible shades with the lines being the lower values and the background as the higher values. In Figure 4b, a convolution filtering process has been performed to enhance the local contrast of the lines against the background. This is commonly referred to as edge enhancement or high-pass filtering, and is well described by Moik (1980). Figure 4c has been converted to a binary valued image by selecting a threshold value that separates lines from background.

Filling Operations

The purpose of a fill operation is to eliminate small voids in a line that add unnecessary complexity. The term fill is also used in computer graphics applications, where it is synonomous with flooding or painting of the interiors of polygons. While somewhat similar, this is not the process we require in processing linework. In the scanning of linework, small voids are sometimes created within the lines as artifacts of the scanning process. In some cases, we wish to eliminate valid inclusionary voids that are too small to be considered significant. In addition to filtering out small voids, filling also tends to generalize and fill in acute angular features on the exterior edges of the linework. This may sometimes be appropriate, especially because subsequent thinning is performed from the line edge inward and is greatly affected by a rough line edge. Figure 3c shows how filling can be used to eliminate small holes or void inclusions in drainage lines that have been extracted from a DEM.

Thinning Operations

Several alternatives exist for thinning or skeletonizing rasterized linework to create a single connected line. Most are iterative, and several use a two-step processing approach. One method performs the bulk of the thinning in a first step with a second step that eliminates all but the single connected lines (Landy and Cohen, 1985). The decision tables for thinning in the first step must not allow thinning to less than two pixels wide. This is to ensure that gaps are not formed by thinning on both sides of a line during the same pass. A similar approach is described by Rosenfeld and Kak (1982) that also requires two steps, and alternately thins first the top and left sides of the linework and then the bottom and right sides, in order to avoid the creation of gaps.

In general, thinning can be performed on all patterns where gaps would not be created by deletion of the center pixel (that is, where 8-way connectivity is maintained), and where the pattern is simple (that is, where only one black-white transition is found as the neighbors are tested in circular order). A special case is posed by the "T" connection (pattern group 16), that will become a "Y" connection, unless given consideration. Also, single or isolated pixels (pattern group 1), may be deleted during thinning, provided they are considered erroneous or insignificant. Endpoints (pattern groups 2-4) may or may not be preserved, as some may be valid lines and some may be spurs that are created as artifacts of thinning of thick lines (more than 4 pixels wide). Spurs may also be deleted after vector conversion, when length may be calculated and used as an editing parameter. Figure 3d shows drainage lines after filling and two-step thinning

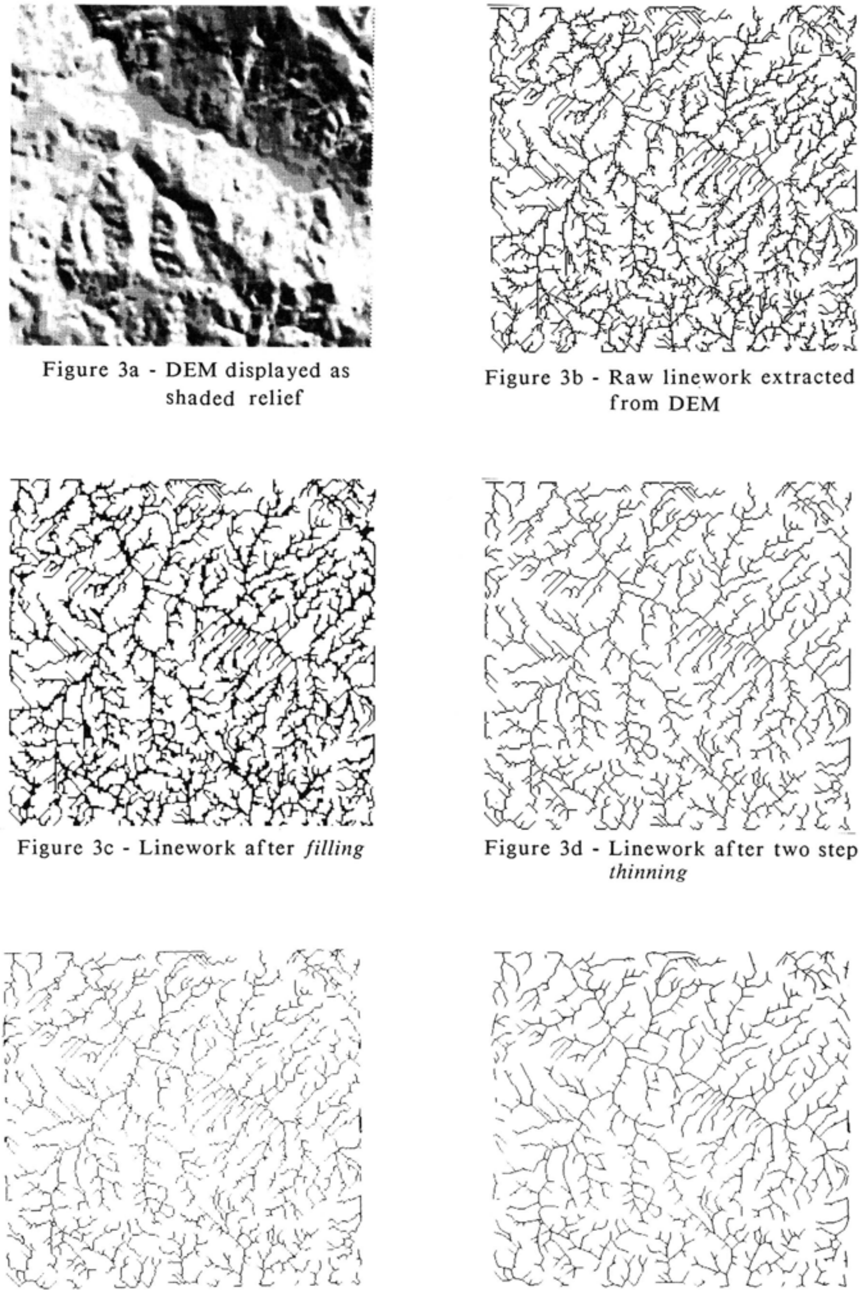


Figure 3a - DEM displayed as shaded relief

Figure 3b - Raw linework extracted from DEM

Figure 3c - Linework after *filling*

Figure 3d - Linework after two step *thinning*

Figure 3e - Linework after conversion to vectors

Figure 3f - Vector linework after *spline* smoothing

Figure 3 - Processing steps for linework extracted from Digital Elevation Model

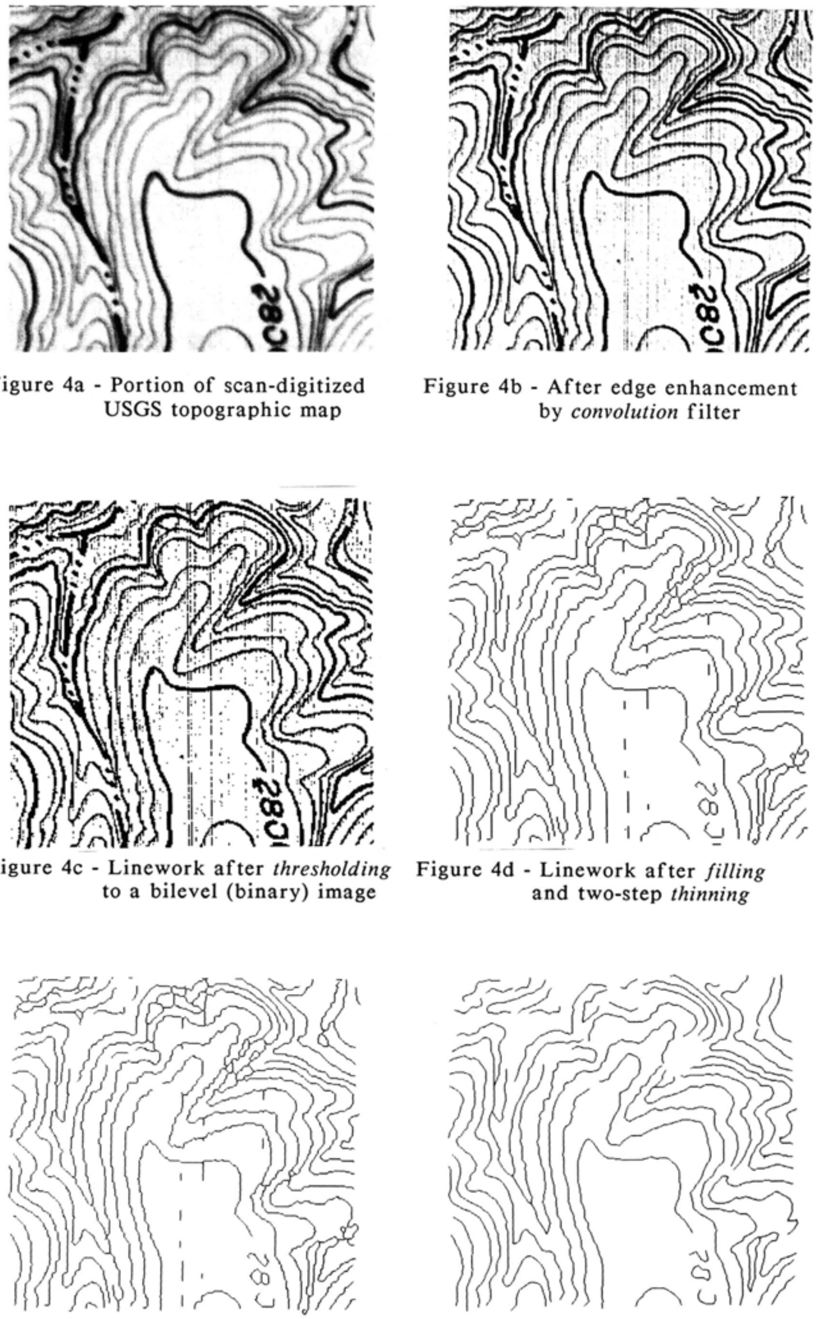


Figure 4a - Portion of scan-digitized USGS topographic map

Figure 4b - After edge enhancement by *convolution* filter

Figure 4c - Linework after *thresholding* to a bilevel (binary) image

Figure 4d - Linework after *filling* and two-step *thinning*

Figure 4d - Linework after conversion to vector format

Figure 4f - Linework after *splining* and deleting short arcs

Figure 4 - Processing steps for scanned topographic linework

have been performed. Figure 4d shows scanned linework after filling and two-step thinning. In both examples, the thinning operation has reduced the original binary image to skeletal connected lines, and has eliminated many unnecessary pixels.

Node Marking

Once lines have been filled and thinned, node marking may be performed. Nodes that are endpoints can be detected as 1-count pixels, or pixels with a single occupied neighbor. Connecting nodes, or nodes where lines join, can be detected as patterns that have three or more transitions, indicating three lines eminating from the node. A special case is the "+" (pattern group 31), that is a connector, even though the gap value is one (that is, no gap would be created if the center pixel were removed). Of special concern is the problem of multiple connecting nodes, or node clusters, that make it difficult to properly place the intersection of lines. More than that, unless reduced to a single point, multiple connectors can create a web of short interconnected arcs to be created during vector generation. Rosenfeld and Kak (1982) suggest that one solution is to perform a second pass through the image, test for adjacent connectors, and calculate a single point (a centroid) that can be used as the connector.

Creation of Digital Line Graph (DLG)

With nodes marked, the process of converting linework into vector arcs can begin. The format for the resulting lines, chains, or arcs may be expressed in relative coordinates (for example, above, above and left), or converted to absolute measure (for example, cartesian coordinates). The USGS Digital Line Graph (DLG) stores absolute coordinates, usually as units of digitizer inches, or as UTM meters. For this project, the DLG was chosen because several available systems could be used to read and write data in this form. Conceptually, the procedure is simple. We begin on an endpoint or connector node and simply follow the skeletonized lines that are guaranteed to have only one path to follow until another node is found. To avoid following a line twice, the pixels are deleted as their coordinates are added to the line. This approach works well when the entire image can be resident in computer main memory, or in an image display memory, as in the case of the system used for this study.

VECTOR OPERATIONS

Point Reduction

Linework that has been converted to vectors is made up of short line segments that have a length of one pixel unit for four-way adjacent neighbors, or of 1.414 units for diagonal neighbors. Straight lines can be reduced by eliminating intermediate points on the line. In addition, lines may be further reduced by using a generalization method such as described by Douglas and Peucker (1973). For the examples used in this study, point reductions of 50-80 percent were possible with little perceivable difference in the quality of the lines.

Line Smoothing

Point reduction does not reduce the appearance of sharp jagged arcs that show remnants of their raster derivation. In fact, point reduction, when used exclusively, will remove blunt angles and leave sharper angles in the lines. Sharp angles that exist at points of inflection can be treated by fitting a curve through the set of points. This will introduce new points but will also yield arcs with a more acceptable appearance. One such technique is cubic spline interpolation and is described by Monmonier (1982). In figure 3f, spline interpolation has been performed on the drainage lines with the result showing a smoother appearance. In figure 4f, spline interpolation has been performed, and in addition, short arcs have been deleted. In this case, short arcs were largely made up of erroneous spurs (generated in thinning) and bridges (between lines).

SUMMARY AND CONCLUSIONS

This study was performed in order to convert rasterized lines to vector arcs in a format appropriate for geographic information system processing. This was accomplished by assembling several image-processing and vector-editing techniques into a prototype system. The documentation of surround conditions may have broader applicability than that required for this project, and it is felt that the use of pattern groups can provide an efficient and consistent framework for many neighborhood operations that are performed in the raster domain. The experiences gained in this project have helped to objectively characterize the steps required in raster-to-vector processing. The understanding of these processes will become more valuable as scanning systems become more affordable and turn-key systems that are tailored to specific requirements become available.

REFERENCES

Douglas, D.H. and Peucker,T.K. (1973), Algorithms for the reduction of the number of points required to represent a digitized line or its caricature: Canadian Cartographer, v.10 , p.112-122.

Golay, M.J.E. (1969), Hexagonal parallel pattern transformations: IEEE Transactions on Computers, v. C-18, no. 8.

Jenson, S.K. (1985), Automated derivation of hydrologic basin characteristics from digital elevation model data: Proceedings of Auto-Carto VII, 1985, Washington, DC.

Landy, M.S., and Cohen, Y. (1985), Vectorgraph coding: efficient coding of line drawings: Computer Graphics and Image Processing, v. 30, p.331-344.

Moik, J.G. (1980), Digital processing of remotely sensed images: NASA SP-431, p.130-140.

Monmonier, M.S. (1982), Computer assisted cartography: Prentice Hall, p. 108-110.

Nichols, D.A. (1981), Conversion of raster coded images to polygonal data structures: Proceedings of Pecora VII Symposium, 1981, Sioux Falls, SD.

Pavlidis, T. (1980), A thinning algorithm for discrete binary images: Computer Graphics and Image Processing, Vol 18, p.142-157.

Pavlidis, T. (1982), Algorithms for graphics and image processing:ˇ Computer Science Press, Chapters 7-9, p.128-214.

Peuquet, D.J. (1981), An examination of techniques for reformatting digital cartographic data, part 1: the raster-to-vector process: Cartographica 18: vol.1, p.34-48.

Rosenfeld, A., and Kak, A.C. (1982), Digital picture processing [Second Ed.]: Academic Press, v. 2, p.232-240.

RECURSIVE APPROXIMATION OF TOPOGRAPHIC DATA USING QUADTREES AND ORTHOGONAL POLYNOMIALS*

Lloyd A. Leifer** and David M. Mark
Department of Geography
State University of New York at Buffalo
Buffalo, New York 14260

ABSTRACT

Orthogonal polynomials (OP) are used to estimate polynomial coefficients and root-mean-square deviations (RMSD) for gridded elevation data within quadtree subquadrants. These subquadrants are recursively subdivided into four if the RMSD exceeds some threshold. Polynomials of orders one through six are fitted to three 256 by 256 DEMs, using RMSD thresholds of 1, 3.5, and 7 meters. The OP-quadtrees required from 9 to 20 percent of original grid space when the RMSD was set at 7 meters, but between 48 and 99 percent of that space for an RMSD of 1 meter. For a fixed RMSD, the total space required appears to be independent of polynomial order. If this effect is true in general, the obvious implication is that order does not matter. In that case, low-order polynomials could be used, saving computation time. When order is held constant, the space required by the OP-quadtree appears to be an inverse power function of the RMSD criterion.

INTRODUCTION

A **digital elevation model** (DEM) can be defined as any machine-readable representation of topographic elevation data. A major issue in DEM research is the selection of an appropriate **data structure** (Mark, 1979). The most frequently-used data structure for DEMs is a regular square grid. One weakness of the grid data structure is its inherent redundancy, and the large amount of computer resources needed to achieve a given accuracy. The grid size must be sufficiently small to capture the smallest feature of interest in the entire study area, and to define the boundaries of larger features to some required level of precision. This implies that cells in most of the region will be smaller than needed; in other words, there will be too many cells.

In an attempt to address this problem, alternative data structures for DEMs have been designed. The most widely used of these is the triangulated irregular network (TIN),

1 This research was supported by National Science Foundation Grant SES-8420789 and SUNY RDF Grant #150-9207-F; the support of both is gratefully acknowledged.

2 Lloyd Leifer currently works for Autometric, Incorporated, 5205 Leesburg Pike, Suite 1308/Skyline 1, Falls Church, Virginia 22041

which represents the terrain by a triangulation on a set of points chosen to represent the surface (Peucker and others, 1978). Whereas TIN are being adopted as a DEM data structure within several of the leading commercial Geographic Information Systems (GIS), the TIN structure is at a disadvantage in terms of data collection. This is because devices are now available to produce very dense regular grids directly from aerial photographs.

At least two strategies for the more direct compression of grid DEMs have been developed. In one approach, grids of varying spatial resolution are employed; in the other, square or rectangular patches of a fixed size are approximated by polynomials or other mathematical functions requiring fewer coefficients than there were grid points. Quadtrees are a spatial data-structure which provides a convenient basis for handling variable resolution data, and allows the variable-resolution and polynomial-patch approaches mentioned above to be combined.

In the present study, orthogonal polynomials are used as an efficient way to estimate polynomial coefficients for gridded data; we call the result the OP-quadtree of the DEM. Polynomials of orders one through six were fitted to each of three 256 by 256 DEMs, using root-mean-squared deviation (RMSD) thresholds of 1, 3.5, and 7 meters. Also, integer RMSDs from 1 to 10 were evaluated for 3rd-order polynomials for the three study areas. Empirical results are presnted, and the implications of these results are discussed.

BACKGROUND

Polynomial Patch Approximations of Topographic Surfaces. Junkins, Jancaitis, and co-workers applied polynomial patch approximations to DEM surfaces (Junkins and others, 1972; Jancaitis, 1977). They were primarily concerned with the handling "noisy" data from the U. S. Army's UNAMACE image correlation system. Their approach was to divide the surface into small square patches, and to fit low-order polynomials (quadric surfaces, $z = a + bx + cy + dxy$) to the elevations within these patches. The patches were small, and the resulting quadric surface was used regardless of how large the residual variance was. (For noisy data, large residuals are presumed to represent errors which should be removed from the data.) Because this method produces discontinuities along patch boundaries, Jancaitis' group used a weighting function approach to blend together adjacent patches, eliminating undesireable breaks in decompressed data.

Quadtrees. The **quadtree** is a data structure which is based on a regular decomposition of a square image into quadrants and subquadrants. Basically, the quadtree can be constructed recursively, with a "stopping criterion" which indicates whether a subquadrant should become a terminal (leaf) node in the quadtree, or should be subdivided. In most quadtree research, the stopping criterion is uniformity, that is, a subquadrant is subdivided if it contains any variation.

There is a major problem with the strict application of quadtree concepts to topographic data: it is unusual to find sets of four mutually-adjacent cells which are of identical height. In order to be space-efficient, the quadtree concept must be adjusted. While the recursive spatial structure of the quadtree is retained, the stopping criterion can be modified to include surface approximation within quadrants (Martin, 1982; Chen and Tobler, 1986). A mathematical function is fitted to the heights within a square subquadrant. Then, whenever the RMSD for the elevations within a subquadrant is larger than some predetermined criterion, the procedure is applied recursively to each subquadrant of the current square.

Quadtree-based Surface Approximation. Martin (1982) used a quadtree-based method for polynomial approximation of DEM data. His procedure fitted a linear equation ($z = a + bx + cy$) to all elevations within a valid quadtree subquadrant. If the RMSD was less than some threshold, the 3 coefficients of the plane were used to represent elevations within the subquadrant. Otherwise, the subquadrant was split into its 4 children, and the procedure was recursively applied to the children. Clearly, the depth of quadtree subdivision, and thus the quantity of data to be stored, will increase if a low threshold is chosen for the RMSD.

In a similar study, Chen and Tobler (1986) fitted five mathematical functions to quadtree subquadrants. The functions chosen were: (1) the mean surface (equivalent to a least-squares polynomial of order zero); (2) a maximum surface (highest elevation in the quadrant); (3) a minimum surface; (4) a "ruled surface," a hyperbolic paraboloid of the form: $z=a_{00}+a_{10}x+a_{01}y+a_{11}xy$; and (5) a quadric surface of the form $z=a_{00}+a_{11}xy+a_{20}x^2+a_{02}y^2$. The coefficients of functions (4) and (5) were determined by substituting into the equation the coordinates of the four corners of the quadrant. Thus the equations pass through the corners exactly, and are not influenced by other cells in the quadrant. Chen and Tobler evaluated goodness of fit according to maximum absolute deviation, rather than RMSD.

The advantage of their approach over a least-squares or orthogonal polynomial method employed in this paper is largely computational efficiency. Furthermore, since the values stored to represent the surface are just elevations, they can be represented using two bytes each; in contrast, polynomial coefficients must generally be represented by floating-point numbers, needing at least 4 bytes each. Chen and Tobler (1986) computed space requirements and running times for two topographic samples, each a 128 by 128 grid of 50 meter cells; they did not discuss the source of their DEM data. Each surface was tested with maximum-error tolerances of 2, 6, and 10 meters. Their test program proceeded recursively, and counted quadtree leaves. The number of leaves was then multiplied by 2 bytes per coefficient plus 2 bytes for the location key (a total of 4 bytes per leaf for functions 1, 2, and 3, and 10 bytes per leaf for 4 and 5). Chen and Tobler found that the 2-meter tolerance produced quadtrees requiring more space than the original grid (2 bytes per elevation) for every function and for each of the topographic samples. For the 10-meter tolerance, the "ruled surface" quadtrees required from 50.7% to 84.4% for the more smooth terrain sample, and from 53.3% to 127.9% for a more rugged area. For every combination of tolerance and topography, the ruled surface required the least space of all functions tested.

Orthogonal Polynomials.
When data are acquired at equally-spaced intervals, orthogonal polynomials provide a computationally efficient way of calculating the coefficients of a polynomial (trend surface) function. Their use in determining least squares coefficients requires substantially less computation because matrix inversion is not required as with traditional regression analysis. As a result, these polynomials have long appeared in **trend surface analysis** (Simpson, 1954; Grant, 1957; for an overview, see Krumbein and Graybill, 1965). One interesting property is that when variables of higher degrees are added to the function, low-order components do not have to be recalculated because the polynomials are independent (hence the term "orthogonal"). For an interesting description of the use of orthogonal polynomials in the one-dimensional case, see Fisher (1973) or Krumbein and Graybill (1965).

For a given sample size N, an N by N matrix, hereafter referred to as the "orthogonal matrix", can be constructed with independent columns that represent individual orthogonal polynomials. If these columns are numbered 0 to N-1, then column i contains the values of the orthogonal polynomial for determining the coefficients of the trend surface variables for degree (exponent) i. Thus, column 0, which is always the unit vector, would be used to calculate the value for a zero-order polynomial equation, resulting in the mean of the dependent variable. In addition, the orthogonal polynomial of column 1 would be used to determine the linear trend in the data; column 2 would be used for the quadratic component. In the same manner, the coefficients for the trend surface function can be computed up to degree N, when a perfect fit is made between the data and the regressive model. DeLury provides a table of these orthogonal matrices up to N=26 (DeLury, 1950). In addition, DeLury also provides a method for generating the orthogonal matrices for larger values of N, but the integral values soon become too large to handle using standard variable types on many computers.

Orthogonal polynomials also can be applied to two-dimensional data. With two dimensions, an N by M data matrix is pre- and post-multiplied by the orthogonal matrices for sample size N and M respectively, thereby allowing the method to be applied to a rectangular grid. However, in the present case, N and M will always be equal, and will be 2 raised to the power of the level. Each value in the resultant matrix, henceforth referred to as a **G**, is divided by the product of the total sum of squares (SOS) for the two orthogonal polynomials that correspond to the value's position in the matrix, producing a new matrix, **B**.

The matrix **B** contains the individual b_{ij} coefficients for the trend surface function, with the position in the matrix indicating the appropriate independent variable; the row represents the exponent of the horizontal (X) component and the column the exponent of the vertical (Y) component. For example, the coefficient for position row=2, column=3 in the **B** matrix would be associated with the independent variable X^2Y^3 of the trend surface equation. It should be noted that the b_{ij} coefficients cannot be interpreted as the marginal effect on the dependent variable as is the case with the traditional b_r regression coefficients. However, the b_{ij} coefficients can be used to determine the appropriate order of the polynomial function, and can be used to generate the best approximate surface for the original elevation data. Krumbein and Graybill (1965) provide an excellent procedural description of the utilization of orthogonal polynomials for a two-dimensional trend surface analysis.

In this study, the criterion for deciding whether a specific polynomial function fits a matrix of data will be based on the square root of the mean square deviation (RMSD) between the polynomial surface and the original data. Computation of the RMSD requires the calculation of matrix **Z** which contains the corrected sum of squares associated with each independent variable. With orthogonal polynomials, the calculation of this matrix is straightforward. First, each element of matrix **G** is squared. Then each value is divided by the product of the SOS of the appropriate orthogonal polynomials in the manner described above. Each value within the resultant **Z** matrix is the corrected sum of squares attributed to an independent variable. Again, the row and column position of a value within the matrix indicates the appropriate independent variable. Given the order of a trend surface function, the sum of all values in the Z matrix whose row and column sum is less than or equal to this order constitutes the corrected SOS attributed to the trend surface equation. The sum of the excluded values reveals the residual SOS, which is divided by the sample size and raised to the 1/2 power to determine the RMSD.

METHODS

As noted above, application of the method of orthogonal polynomials to a grid of N by N cells produces a matrix of N^2 coefficients from which the surface can be recreated perfectly (except for truncation and roundoff errors resulting from machine representations). The quadtree-based algorithm computes the orthogonal polynomial coefficients (OPC) for a quadrant, and then uses a decision rule to determine whether the current quadrant should be saved or whether it should be recursively subdivided.

Various strategies for selecting a subset of the OPC, and for assessing whether the resulting fit is 'adequate', lead to different algorithms. In the current study, we used a fixed polynomial order for each run. Because of limitations on computer time and space, the orthogonal matrix for N=32 could not be generated; thus, in our experiments, the quadtree subdivision begins with level-4 nodes (16 by 16 subgrids). Then, the SOS terms associated with all OPC with exponents less than or equal to the polynomial order are added together and used to compute residual RMSD. If this RMSD is greater than the specified maximum allowable error, the data matrix is divided into four subquadrants, and the fitting procedure is applied recursively to each of these. The program does not subdivide the data matrix if the RMSD is less that the threshold, or if the data matrix is too small to calculate a surface function of the specified order.

As noted above, a polynomial trend surface of order O is the sum of polynomial terms of the form $a_{ij}x^iy^j$, such that $i+j \leq O$. It can easily be shown that the total number of terms in such a polynomial is equal to [(O+1)(O+2)]/2. This is the number of degrees of freedom available for least-squares estimation of the trend surface, and also represents the minimum number of observations needed to compute the polynomial. However, in order to use the method of orthogonal polynomials, a matrix of size (O+1) by (O+1) must be computed, requiring a data grid of side-length O+1. This places a more severe restriction on the possible polynomial orders which can be computed for small matrices: the maximum order is one less than the side length of the data square. Table 1 relates quadtree levels, space requirements, and possible polynomial orders.

TABLE 1: QUADTREES PROPERTIES AND ORTHOGONAL POLYNOMIAL STATISTICS

1) quadtree properties:

quadrant level	0	1	2	3	4	5
pixels	1	4	16	64	256	1024
bytes	2	8	32	128	512	2048

2) orthogonal polynomial properties:

order	1	2	3	4	5	6
OP terms	3	6	10	15	21	28
bytes for OPC*	14	26	42	62	86	114
minimum level**	2	2	3	3	3	3

* calculations assume that each coefficient is stored as a floating-point number requiring 4 bytes, and add 2 additional bytes for the location key

** minimum level for which OPC require fewer bytes of storage than the original integer grid

We noted above that Chen and Tobler (1986) counted the total number of quadtree leaves of any size (level) needed to represent the surface to some pre-determined accuracy, and then multiplied that number by the number of bytes needed to represent one quadrant of the current equation. For some surfaces and tolerance values, Chen and Tobler found that some quadtrees required more storage space than the original grids. However, it seems more appropriate to store a hybrid structure, counting a leaf-node only when the surface function requires less space than would the original grid over the same quadrant. In the present study, we did not subdivide a subquadrant if the OPC for its four children would require more bytes of computer storage than would the integer elevations of the grid cells within the patch. Minimum space-efficient levels for each order of OP are given in Table 1. Because our objective was to examine space-efficiency, our program simply counted the number of leaves of each level which are needed to represent the surface; in an actual application, the appropriate elements of the coefficients matrix **B** would be stored along with a key-number denoting the location of the quadrant, either in a pointer-based quadtree or a linear quadtree.

TERRAIN SAMPLES

We applied the methods discussed above to three topographic samples. Each sample was a 256 by 256 sub-sample of a U. S. Geological Survey 7 1/2 minute digital elevation model (Elassal and Caruso, 1983). The DEMs were collected as by-products of orthophoto quadrangle mapping, and all three have a grid cell size of 30 meters and a vertical height resolution of 1 meter. The two Pennsylvania samples were collected using a Gestalt Photo-Mapper II (or GPM-II; Swann and others, 1978; Elassal and Caruso, 1983); the sample from Oregon was produced using a semi-automatic B-8 stereo-plotter (Elassal and Caruso, 1983). Possible effects of data collection methods on DEM characteristics were discussed by O'Neill and Mark (1985).

The three test areas represent distinctly different types of topography, and have been used in previous DEM studies of fractals (Mark and Aronson, 1984) and topographic slope (O'Neill and Mark, 1985). The Blair's Mills (Pa) quadrangle is located in the Appalachian Mountains. Strong structural control has produced a series of aligned ridges and valleys oriented in a northeast-southwest direction. In contrast, the Keating Summit (Pa) quadrangle represents topography developed in the flat-lying sedimentary rocks of the Appalachian Plateau, showing little if any sign of structural control. Finally, the Adel (Ore) quadrangle is from the Basin-and-Range topographic province in south-eastern Oregon. In this area, the steep fault scarps and associated canyons contrast sharply with the gently-sloping plateau above and the flat valley floor below.

RESULTS

As noted above, the program we used began by breaking each 256 by 256 grid into 16 by 16 (level 4) subgrids. Then, for each of these, a surface of the current order was fitted to the data; if the RMSD was larger than the current threshold, the square was divided into four 8 by 8 squares, and the procedure was applied recursively. Table 2 presents the main results of the 54 runs of the program (3 DEMs times 3 RMSD thresholds times 6 polynomial orders).

As expected, for each order and for each DEM, the space required declines as the RMSD threshold increases. Unexpected was the apparent independence of space required and polynomial order (for fixed RMSD and DEM): each column in Table 2 contains values that are relatively constant. (We do, however, note that third-order polynomials were best in 4 cases, whereas first- and sixth-order polynomials were never the most space-efficient.) The fact that space-efficiency is almost independent of polynomial order suggests that, for a given RMSD threshold, the DEM has a fixed "information content", which can be expressed in bytes. The DEM can be approximated by many small patches containing simple surface polynomials, or by a smaller number of more complicated surfaces, and the two effects seem to cancel out. This should be the subject of further investigation.

The other pattern evident in Table 2 is that, for the 1-meter RMSD threshold, the Adel sample required far less space than the others. This might be in part due to the nature of the terrain, which (as noted above) consists of fault scarps, flat valley floors, and fault dip-slopes in the form of inclined planes. However, we believe that the difference is primarily due to the short-scale error characteristics of the data, which are chiefly dependent upon the method used to collect the data. The Gestalt

TABLE 2: SPACE REQUIREMENTS* FOR THE OP-QUADTREES
FOR POLYNOMIAL ORDERS 1 THRU 6

Quadrangle:	Adel			Keating Summit			Blair's Mills		
RMSD:	1.0	3.5	7.0	1.0	3.5	7.0	1.0	3.5	7.0
1	57.7	24.6	11.3	96.3	37.6	16.3	96.6	38.3	13.2
2	53.3	20.3	10.0	90.8	28.8	12.9	93.9	36.7	11.3
3	53.6	19.7	9.5	99.3	28.0	9.6	98.4	30.8	11.5
4	48.0	19.8	12.4	94.5	24.5	12.3	97.4	32.8	12.5
5	50.0	21.3	16.8	88.5	20.2	16.8	88.7	33.2	16.8
6	60.3	24.1	22.3	92.2	23.6	22.3	94.7	31.9	22.3

* figures in table are required space as a percentage of space required by the original grids

Photo-Mapper (GPM-II) collects what is essentially a tree-top surface. The model contains error with a magnitude of about half the average tree height, and with a local structure similar to white noise. Such uncorrelated errors make it almost impossible for simple polynomials to approximate local areas when the maximum RMSD is set at 1 meter. The effect declines with increasing RMSD (see also Table 3, below). We expect that similar results would apply for other DEM data collected using the GPM-II.

In order to provide a more detailed evaluation of the interaction between maximum RMSD and space-efficiency, we fitted third-order polynomials to each of the three DEM samples using RMSD thresholds ranging from 1 to 10 meters; the results of these evaluations are presented in Table 3. The relation between space and RMSD for each DEM appears to be well-approximated by a power function (straight line on log-log graph paper). These curves could be compared to those generated for other DEM data structures, such as TINs prepared to approximate grids to within some pre-defined tolerance.

TABLE 3: SPACE REQUIREMENTS* FOR THIRD ORDER POLYNOMIALS

maximum RMSD (m)	Adel	Blair's Mills	Keating Summit
1	53.6%	98.4%	99.3%
2	31.1%	74.9%	54.1%
3	22.2%	39.6%	31.9%
4	16.9%	25.0%	24.4%
5	14.0%	17.9%	17.4%
6	11.2%	14.2%	12.1%
7	9.5%	11.5%	9.6%
8	9.1%	10.0%	8.7%
9	8.5%	9.1%	8.3%
10	8.2%	8.6%	8.2%

* figures in table are required space as a percentage of space required by the original grids

SUMMARY

For a fixed RMSD, the space requirements appear to be relatively independent of polynomial order. High-order polynomials fit large regions more easily, and thus the OP-quadtrees have fewer leaves; however, each leaf requires more space for the polynomial coefficients, and the two effects seem to cancel. If this can be shown to be true in general, the implication is that the order used does not matter. In that case, low-order polynomials should be used, since they can be computed more efficiently. When order is held constant, space requirements for the OP-quadtree appear to be an inverse power function of the RMSD criterion for low values of it (RMSD < 10 meters).

REFERENCES

Chen, Z.-T., and Tobler, W., 1986, Quadtree representations of digital terrain: Proceedings, Auto Carto London, v. 1.

DeLury, D. B., 1950, Values and Integrals of the Orthogonal Polynomials Up to n=26: Toronto, University of Toronto Press.

Elassal, A. A., and Caruso, V. M., 1983, Digital elevation models: U.S. Geological Survey Circular 895-B, 40pp.

Fisher, R. A., 1973, Statistical Methods for Research Workers: 14th Edition, New York, Hafner Publishing.

Grant, F., 1957, A problem in the analysis of geophysical data: Geophysics, 22, 309-44.

Jancaitis, J. R., 1977, Elevation data compaction by polynomial modelling: ASP-ACSM Joint Annual Spring Convention, Washington D.C., Spring, 1977.

Junkins, J. L., Miller, G. W., and Jancaitis, J. R., 1972, A weighting function approach to modelling of geodetic surfaces: Journal of Geophysical Research.

Krumbein, W. C. and Graybill, F. A., 1965, An Introduction to Statistical Models in Geology: New York, McGraw-Hill.

Mark, D. M., 1979, Phenomenon-based data-structuring and digital terrain modelling: Geo-Processing, 1, 27-36.

Mark, D. M., and Aronson, P. B., 1984, Scale-dependent fractal dimensions of topographic surfaces: An empirical investigation, with applications in geomorphology and computer mapping: Mathematical Geology, 16, 671-683.

Martin, J. J., 1982, Organization of geographic data with quad trees and least squares approximation: Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, Las Vegas, Nevada, 458-463.

O'Neill, M. P., and Mark, D. M., 1985, The use of digital elevation models in slope frequency analysis: Proceedings, Sixteenth Annual Pittsburgh Conference on Modeling and Simulation, 16(1), 311-315.

Peucker, T. K., Fowler, R. F., Little, J. J., and Mark, D. M., 1978. The triangulated irregular network: Proceedings, Digital Terrain Models (DTM) Symposium, ASP-ACSM, St.Louis, Missouri, May 9-11, 1978, 516-540.

Simpson, S. M., 1954, Least squares polynomial fitting to gravitational data and density plotting by digital computers: Geophysics, 19, 255-70.

Swann, R., Thompson, J., and Daykin, S. E., 1978, Application of low cost dense digital terrain models: Proceedings, Digital Terrain Models (DTM) Symposium, ASP-ACSM, St.Louis, Missouri, May 9-11, 1978, 141-155.

CARTOGRAPHIC DATA ENTRY THROUGH AUTOMATIC FEATURE TRACKING

K. Stuart Shea
PAR Government Systems Corporation
1840 Michael Faraday Drive, Suite 300
Reston, Virginia 22090

ABSTRACT

Many cartographic systems currently rely on raster-scan digitizing to convert analog source material to digital form. Raster technology is very effective at rapid and accurate digitization of large volumes of cartographic data. At the same time, existing raster-to-vector (R-V) conversion processes rely on an inordinately large amount of human post-scan editing to coherently sort and combine the short, unattributed lineal segments into single cartographic spatial entities. The Automatic Feature Tracking (AFT) system addresses one of the major causes of this bottleneck in the digitization process—skeletonization. This is accomplished by **directly** converting *symbolized* linear features on a raster map image into sets of *x,y* coordinates. The system relies on Template Matching and Feature Tracking techniques to locate feature centerlines. This paper briefly reviews the history of map digitization techniques, illustrates inadequacies of those past approaches, and presents the AFT system as an alternative to the R-V conversion routines in existing raster digitization systems.

INTRODUCTION

The Evolution of Map Digitization

Since its inception in the early 1960's, digitization processes have seen many technological advances (Boyle, 1979, 1982). Many innovative techniques have been explored and expensive hardware developed to aid in the conversion of hardcopy graphics to digital form. Cartographers have witnessed this technological transition first-hand (Penney, 1979). Early cartographic digitization efforts began by paralleling the traditional use of the information in drafting vectors. Preservation of the vector-nature of the data functionally was similar to intuitive cartographic production processes. Although originally collected by hand, the manual digitization task was so laborious that special digitizing equipment was soon developed. Over the last 25 years, the resulting equipment—manual digitizers—has evolved considerably. From the mechanical, arm-type digitizers of the early 1960's, to present-day electromagnetic induction free-cursor digitizers, this maturation continues.

As the need for more types of digital cartographic data emerged, practitioners were also experiencing a growing dependency on the digital information. It soon became obvious that the existing manual digitization techniques could no longer support the requirements of the cartographic community. The late 1960's and early 1970's saw a movement away from vector-oriented data collection systems, and significant attention was placed upon raster digitization. Initial application of raster data to cartography was limited to the area of production; specifically, to the generation of color film separates. As a means for *mass digitization*, though, the benefits were much more obvious. Cartographers agreed with alacrity that raster scanning was the preferred method of data capture for the future.

Nonetheless, raster scan digitization met with many obstacles: 1) the very nature of raster data violated the traditional mind-set of cartography being a vector-based process; 2) features were no longer identified as discrete elements; 3) vector, coordinate geometry representation for data storage was more efficient; and 4) existing manipulative processes continued to be vector-oriented (Peuquet, 1979, 1982). Despite these apparent shortcomings, the raster data capture movement persisted and continues to date. This is primarily borne of necessity because: 1) raster data collection speeds far exceed those of manual digitization; 2) it generally removes the operator from the conversion loop, thereby eliminating humans frailties in the digitization process (such as physiological and psychological deficiencies); and 3) achievable accuracy is much greater. As a result, raster scanning is now finding its way into many mapping producers' mass digitization systems.

Shortcomings in Existing Raster-to-Vector Conversion Techniques

Until such time as raster data can be efficiently stored, manipulated, and exploited in the production process, the major remaining obstacle in its outright acceptance as the preferred method of data capture exists in the conversion of the scanned data into the familiar spatial format of vector-based lineal chains (Fegeas, 1983). Raster-to-Vector (R-V) conversion processes are numerous and have been well defined in the literature. Peuquet (1981) suggests that a generic R-V process can be divided into three basic operations: 1) skeletonization; 2) line extraction, or vectorization; and 3) topology reconstruction. A major disadvantage of existing R-V conversion routines is that they do, in fact, follow these steps, and do not effectively exploit the inherent symbolic nature of the cartographic features in the raster map image. This is partly due to the fact that past R-V conversion efforts have primarily focused on routine cartographic symbolized features (such as primary/secondary roads, contours and polygons)—those with simple linework—and have generally ignored those problems dealing with complex, highly stylized symbolized lines (such as railroads, cased roads, trails, and intermittent drains).

Existing maps contain a wealth of information which is ignored and eventually eliminated in most R-V conversion techniques. It would be more advantageous to eliminate the skeletonization process, and vectorize the raster data by exploiting the symbolic nature of each feature to discriminate it from its surroundings. By operating on symbolized features instead of skeletonized features, this direct conversion would also eliminate a number of weaknesses which result from other skeletonization techniques. Consider, for example, the flaws which result from converting the following feature types:

- *Railroads.* Railroad features are converted to a dash-tick-dash-tick representation wherein each dash and perpendicular tick would be separate vectorized features. An operator is required to combine these numerous segments into a single entity and tag it as a single railroad feature.

- *Cased Roads.* The actual feature location is the center of the casing yet, typically, skeletonization routines capture the two "parallel" lines as distinct features. These two distinct lines must subsequently be tied together to represent a single cased road.

- *Intermittent Drains.* The dash-dot-dot-dot-dash symbology sequence for this type of feature complicates normal R-V conversion. The dots are normally discounted as noise in the skeletonization process, and the dashes end up as individual linear features requiring subsequent tagging as a single drainage feature.

Besides the problems that arise due to different symbol types, many anomalies found on paper maps contribute to the degradation of digital data quality during the digitization process. Manual lineal digitizing systems, as well as typical R-V conversion systems, have not overcome the influence of these anomalies and, in fact, magnify their impact on the integrity of the data collected. For example, the following inadequacies exist in standard skeletonization routines:

- The wholesale skeletonization of an entire data file, in which all features are converted to a one (1) pixel width, typically causes the generation of stubs, gaps, and unthinned data elements.

- Pre-processing the source documents, such as in the creation of film positives for scanning, may be required and involves a significant amount of effort (Antell, 1983).

- The algorithms are highly sensitive to local variations in line width or breaks/gaps in line and will influence the positional accuracies of vectorization (Selden, 1986).

- Scanning noise caused by blemished manuscripts, or folds in the original source, will adversely affect the conversion.

Future developments of advanced mass digitization systems must be able to meet the growing need for, and reliance on, digital cartographic data. The digitization process must be accomplished with limited human intervention while exploiting the rapid data entry capabilities of raster scanning. It is obvious that the current R-V conversion systems are failing in their claims of reducing the human bottleneck; too much effort is often required in subsequent editing/cleanup tasks to make the systems viable. PAR Government Systems Corporation (PGSC) has identified feasible solutions, and has implemented streamlined techniques to minimize, if not eliminate a major cause of the digitization bottleneck. The integration of these techniques into a cohesive testbed for enhanced feature discrimination forms the basis of the Automatic Feature Tracking (AFT) system.

AUTOMATIC FEATURE TRACKING

AFT System Overview

The AFT algorithm uses Template Matching and Feature Tracking techniques to locate feature centerlines in a raster map image. Each of these important aspects of the system is described below.

Template Matching. The template matching technique is the key to the AFT algorithm. A raster template is compared to a feature and, if the two match, a point is saved to describe the feature's location. The AFT system operates on raster-scanned binary images in which the features are represented by ones, and the background by zeroes. A template matching algorithm that exploits this parity of values operates by comparing templates with the same values (of 0 or 1) to the map image. Thus, the template matching technique compares a template array to a subarray of the map image. If the corresponding pixels of the template and the map subimage are nearly identical, the correlation between the two is high, and the template location is considered a match for a coordinate depicting the feature.

Binary template matching, however, is not perfect. Features do not generally align themselves with the orthogonal nature of raster data. Linear features represented in raster format often have jagged, stair-step edges. These jagged edges are irregular and unpredictable. Similarly, in many cases the line width of the feature varies because of scanning noise, drafting inconsistencies, and damage to the original manuscript. In an attempt to bypass these problems, a ternary template matching technique provides more flexibility. The ternary template array consists of three values: 1) ones (feature of interest); 2) zeroes (background); and 3) twos (a special value). To accommodate for feature boundaries which are not perfectly parallel and not of uniform width, the neutral (2s), or "don't care" zones, do not contribute to the correlation calculation used by the template matching algorithm.

Extensive experimentation has been conducted to determine the appropriate template design for a number of lineal features on a variety of raster-scanned source materials. Figure 1, for example, illustrates the ternary template matching technique with a template designed for a dashed line.

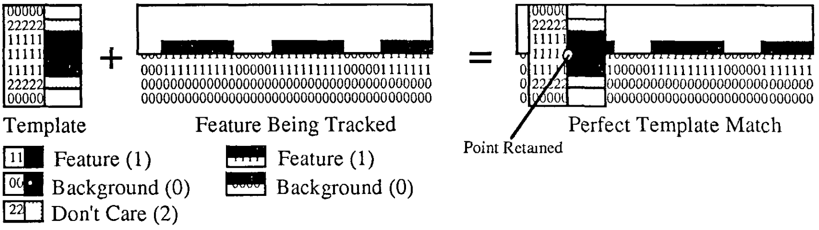


Figure 1. **Ternary Template Matching.** AFT fits a square or rectangular template to match the inherent feature characteristics in a raster map image. Here, the 2s compensate for the intermittent and highly sinuous nature of dashed lines.

Template design parameters are based upon a variety of feature characteristics and differ considerably between feature type. Templates are created to match the physical characteristics of the feature (such as the width or degree of sinuosity). These templates are then iteratively optimized by experimentation to compensate for tracking failures which may result from sharp turns, obstacles, feature look (dashes, dots, crossings), and feature continuity. This process continues until the designed template successfully tracks the feature in question. The template can then be used for subsequent tracking on the same product.

When the appropriate tracking template is placed over a subarray of the map image, a correlation procedure determines the *best fit* of the template to the feature. The template is rotated to check the correlation at many orientations. This allows the algorithm to detect changes in the alignment of the feature. A file of pre-rotated templates is created in advance in order to eliminate the need to perform template array rotations during template matching and tracking. While matching is in progress, the current feature orientation is determined by comparing a subset of these pre-rotated templates to the feature. The orientation with the highest correlation value is chosen.*

In many cases, when the template is initially placed on the feature for comparison, it can be placed 2-3 pixels off-center, and an inappropriately low correlation value may be result. To compensate for this problem, a template matching search window is used. This search window consists of an array of locations where the template will be compared, resulting in only a slight corrective procedure for template centering. Combined with template rotation, this process allows the algorithm to compensate for small changes in the location and alignment of the feature.

Feature Tracking. Once an adequate template-to-feature match is initially obtained in a local search window, the second major part of the AFT system, feature tracking, takes over. Here, the current orientation of the template and a user-specified projection distance is used to predict the next position in which to continue the template matching procedure. Figure 2 illustrates the basic algorithm approach for feature tracking.

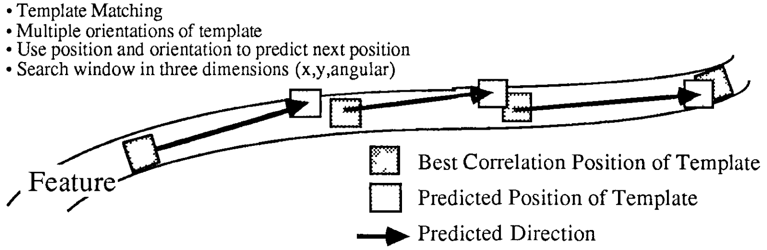

• Template Matching
• Multiple orientations of template
• Use position and orientation to predict next position
• Search window in three dimensions (x,y,angular)
Feature
Best Correlation Position of Template
Predicted Position of Template
Predicted Direction

Figure 2. **AFT Feature Tracking Approach**. The template matching algorithm selects a predicted position based upon the current position and orientation of the template. The template is projected down the feature and the template is rotated and translated until a sufficiently high correlation of template-to-feature is achieved. This location now becomes the new anchor position for the subsequent projection.

The main algorithm used during feature tracking is known as **AutoProject (AP)**. This process governs the template matching algorithm by providing the locations in which there exists a high degree of probability that a feature will be present based upon past experience. AP uses known characteristics of the feature (such as degree of sinuosity or maximum radius of curvature at the source's scale) to predict a new location of the feature being tracked some n-pixels away from the current position. The AP algorithm is illustrated in

*The parameter files associated with the AFT system allow one to specify a *sufficiently acceptable figure of merit* (correlation) which will not require the "highest" correlation.

Figure 3. The AP process is repeated until an individual feature is tracked from a user-selected starting point to ending point.

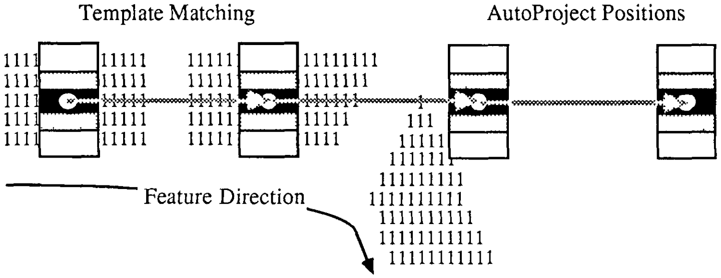


Figure 3. **AutoProject Algorithm**. In this illustration, the template has been projected along a feature. The initial AP has located the feature. Two subsequent projections, though, have failed because of a change in feature direction.

When there are features with gaps, such as dashed lines, or features which have a high degree of complexity/sinuosity, such as intermittent drains, two variations on the AP algorithm control the tracking. The **Increased Area Search (IAS)** and **Center of Mass (CM)** algorithms are used to jump over gaps and locate features when an adequate template-to-feature match cannot be found during the normal AP sequence. The feature location program control logic dictates whether or not these secondary algorithms are invoked, depending on a set of pre-defined parameters. These parameters also include include algorithm-specific control values such as the initial and secondary projection distances, angular "look" of the projection, and required correlation values.

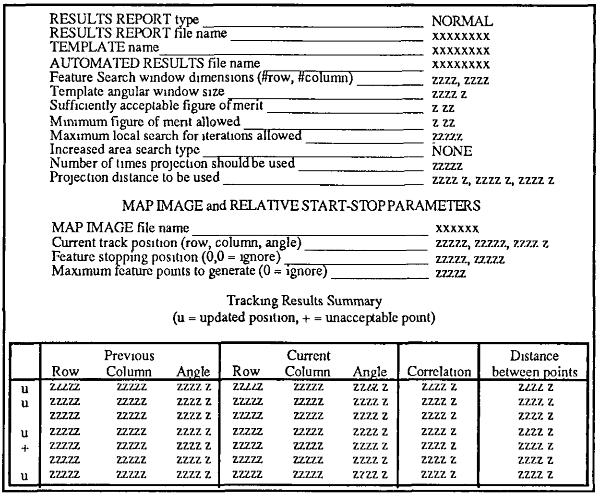
AFT System Design

The AFT system is divided into two main program categories: **feature tracking** programs and **map template utility** programs.* The heart of the feature tracking programs is the feature location program, and controls the logic used while tracking linear features.

AFT System Operations Concept. A typical operational mode using the feature tracking system involves: 1) displaying a portion of a raster map image on an interactive graphic display device; 2) obtaining a starting point on the desired feature to be tracked (using a graphic input device); 3) capturing a point from the display to indicate the direction in which the feature is to be tracked; 4) entering the type of linear feature that is being discriminated (such as a railroad or trail); and 5) running the feature location program.

AFT Tracking System Output. Tracking of the features commences from the user-specified starting point to the stopping point (if one was specified), or until the algorithm reaches the end of the feature. Output from the feature tracking software includes a summary of the tracking results (*Results Report*) and a file containing the actual tracked coordinates (*Automated Results*). Figure 4 is a sample of the report format used in the AFT system.

*The AFT system consists of 11 main programs, 130 subroutines, 20 command files, and 70 other required support files, totalling approximately 12,000 lines of executable FORTRAN 77 code. This does not include any device drivers or graphic interface packages.



RESULTS REPORT type ______________________ NORMAL
RESULTS REPORT file name ______________________ xxxxxxxx
TEMPLATE name ______________________ xxxxxxxx
AUTOMATED RESULTS file name ______________________ xxxxxxxx
Feature Search window dimensions (#row, #column) __________ zzzz, zzzz
Template angular window size ______________________ zzzz z
Sufficiently acceptable figure of merit ______________________ z zz
Minimum figure of merit allowed ______________________ z zz
Maximum local search for iterations allowed ______________ zzzzz
Increased area search type ______________________ NONE
Number of times projection should be used ______________ zzzzz
Projection distance to be used ______________________ zzzz z, zzzz z, zzzz z

MAP IMAGE and RELATIVE START-STOP PARAMETERS

MAP IMAGE file name ______________________ xxxxxx
Current track position (row, column, angle) ______________ zzzzz, zzzzz, zzzz z
Feature stopping position (0,0 = ignore) ______________ zzzzz, zzzzz
Maximum feature points to generate (0 = ignore) __________ zzzzz

Tracking Results Summary
(u = updated position, + = unacceptable point)

		Previous			Current			Distance
	Row	Column	Angle	Row	Column	Angle	Correlation	between points
u	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
u	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
u	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
+	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z
u	zzzzz	zzzzz	zzzz z	zzzzz	zzzzz	zzzz z	zzzz z	zzzz z

Figure 4. **Sample NORMAL RESULTS REPORT Format.**

Contemporaneous Tracking Results Viewing. Management of the graphics display is an important operational aspect of the AFT system and is controlled by a separate program. During the feature tracking process, the correct map view display is automatically maintained and the digitized coordinates are plotted over the raster representation of the feature being tracked while the feature location program continues. These two processes communicate with each other through a shared common storage region and through operating system-supported event flags which alert each process to its counterpart's state. When tracking begins, an image view monitor plots track coordinates on the current map view. As soon as a track coordinate gets close to the edge of the current display window the feature location program will automatically display the next map view so that tracking progress can continue to be monitored. As tracking of a feature nears this *trigger region*, a new image view window is displayed with at least 1/4th of the previous image view window retained for reference purposes.

Post-Tracking Results Viewing. In certain cases, it is desirable to view a previously created track. This is especially useful if the user has not requested visual monitoring of the track coordinates during the initial tracking process or if tracking was performed as a batch operation. Many viewing options are available including displaying multiple tracks on the same map view, or that of viewing a long, continuous track.

System Parameter Files. The AFT system was designed to be a modular, system-independent utility. As such, most of the controlling logic for the software is governed by parameter files. These files enable the system to be modified for: 1) new products and features; 2) multiple scan resolutions; and 3) variegated human interaction preferences in the operation of the tracking programs. Parameter files are usually created prior to running the feature location program. The following are examples of AFT parameter files:

- *Current Image View Parameters* file indicates the current or latest map image file name and the area in which tracking is occurring.

- *Relative Starting/Stopping Parameters* file contains the feature tracking start point, the initial tracking direction, and an optional stopping point.

- *Restart Parameters* file contains all information required to reinitiate tracking when the system is interrupted from processing by user request or operating system failure. This file is automatically created each time tracking is initiated. If the track being monitored is not following the intended path, the operator may abort the tracking and manually traverse the trouble spots on a failed tracking attempt. When tracking is reinitiated, it will automatically access the restart parameters and continue tracking.

- *Operation Parameters* file controls certain operational functions of the feature location program; for example, whether or not to graphically display the tracked coordinates during feature tracking.

- *Feature Parameters* files contain the parameters which indicate the methods used to track a particular type of linear symbology. A different feature parameters file is normally created for each type of feature to be tracked.* The *Feature Parameter* files are one of the most important aspects of the AFT system. If a feature parameters file does not exist, the feature parameters must be manually entered. The system also supports a *training* capacity mode—such as when a new product or feature type is being tracked for the first time—wherein parameters are omitted from the feature parameters file and replaced by an "ask operator" signal value. This allows that particular parameter to be fine-tuned through repetitive adjustment since the operator will be queried for that parameter each time the feature location program is invoked. It should be noted, though, that operator interaction is the exception; in a typical operational scenario, the feature parameters are stored for automated use.

AFT System Summary

The AFT system has demonstrated the ability to track a variety of symbolized linear feature types (railroads, cased roads, dashed lines, intermittent streams, and cut-and-fill contours) as well as simple linework (index, intermediate, and supplementary contours). Table 1 provides a sample timing summary from a number of tracking experiments conducted with the AFT system.

Feature Type	CPU Time
Railroad	12 seconds
Cased Road	13 seconds
Dashed Line	16 seconds
Intermittent Stream	11 seconds
Cut & Fill Contour	14 seconds
Index Contour	12 seconds
Supplementary Contour	16 seconds
Intermediate Contour	11 seconds

Table 1. **AFT Tracking Timing Summary**. This table illustrates the amount of time required to track a two-inch segment of each feature type in a batch-mode environment on a time-shared VAX-11/780 system.

It is important to realize that these timing results represent the conversion of a single feature running throughout a complex raster map image with the dimensions of approximately 25,000 by 40,000 pixels. Thus, vector representations of each of these features were collected without a wholesale vectorization of the entire map image, and without the need to undergo post-scan editing to remove noise. The quality of the vectorized line data from the feature tracking process is also exceptional. Most R-V conversion or manual digitization standards require that centerlines be collected to within 1/2 of a line width from the centerline of the feature originally being digitized.The tracking results consistently exceeded that tolerance.

*These files have proven to be very robust for a given type of feature, regardless of the map or chart source. PGSC has designed feature parameter files which have worked on a number of products, with each product originally being scanned on different scanners.

Significance of the AFT Technology

The production programs of the major map and chart producers in the U.S.—the United States Geological Survey (USGS), the Defense Mapping Agency (DMA), and the National Ocean Survey (NOS)—are in a state of transition to all-digital, softcopy production capabilities. This transition includes the establishment of uniform procedures relating to the collection, screening, evaluation, editing, symbolization, retrieval, and exchange of digital source and production data (Franklin and Holmes, 1978). As part of this move to the digital mode, significant efforts are being made to expand and improve mass digitization capabilities (Starr, 1986; Callahan and Broome, 1984). These new capabilities will support the population of multi-product, multi-purpose digital cartographic data bases currently under design. The ability to support a growing dependency on digital cartographic data, coupled with a requirement to meet, or exceed, existing collection system accuracies, will require a rethinking in digitization techniques.

The AFT development is clearly aligned with the current trends at the USGS, DMA, and NOS production centers, and surpasses their accuracy requirements for data collection. The significance of AFT's capabilities are important because of the following:

- Processing of Degraded and Highly Variant Source Materials. Map producing agencies will continue to use a wide variety of hardcopy source materials for inclusion in their digital data bases. Significant among them are the maps and charts produced by and of foreign nations. These maps incorporate a wide variety of symbology schemes consisting of unique line thicknesses, line configurations (regular dashes, irregular dashes, dashes and dots, etc.), and screened lines. In many cases, unsophisticated graphic arts techniques during map production render a wide variation in line quality within the frame of a single map sheet. Infrequent occurrences of damaged maps with stains, pronounced fold marks, and other detrimental effects cause special problems for map conversion activities. Many types of skeletonization and vectorization methods fail or suffer severe throughput degradation when processing damaged or foreign products. The AFT system's unique method of template configuration with non-correlated pixels within the mask effectively accommodates the variances in line quality and performance is not significantly degraded by poorly handled, damaged source materials. In addition, the operator can quickly design and build the correctly configured template along with the modified parameters to allow the data to be processed.

- Discrete Feature Selection. As maps and charts are converted and put into the proposed all-digital production flows, every method of conserving computation power and local data storage becomes significant. Many types of skeletonization and vectorization techniques convert all data within the map frame into vector format. Many of the vectorized features are eventually discarded. This problem represents a significant cost factor in an all-digital production environment. The AFT system's approach avoids the wholesale processing of all the data within the map frame by allowing the operator to select only the desired features for processing and extraction from the map sheet. Savings in processing requirements alone can easily provide a 50% increase in throughput rates.

- Batch Operation. Processing and storage resources are not the only areas where AFT can make a significant contribution to cost control. Future labor savings, perhaps a map producing agency's most costly resource, are of key interest to production planners. Map conversion via manual techniques, that is, table digitizing, is not a cost-effective method of using labor. AFT provides the capability for the operator to enter only the starting position and direction of the feature of interest. AFT does the rest. Experiments have been conducted with a variety of features using AFT in a batch mode and the success rate clearly identifies AFT as a time-saving and labor-saving device.

FUTURE AFT DEVELOPMENTS

System Improvements

The AFT system is by no means static. With the ever-changing digital cartographic production environments at the major map producing agencies, the AFT system remains a dynamic utility. As part of this evolution, many topics are being addressed in future versions including:

- *Feature Parameter Library*. Product-dependent digitization will be supported for a variety of sources (such as DMA's TM-50; USGS's 1:24,000-scale quadrangle maps; and NOS's Nautical and Bathymetric Charts). Feature parameter files, operation parameter files, and templates files are being designed for each product. New parameters and templates can then be added to incorporate new and non-standard map/chart specifications.

- *Automatic Template Generation*. A semi-automated, rule-based approach is being considered to create feature tracking templates. One proposed solution is to have a template creation program query the operator for both general and specific characteristics of the feature. This information includes: 1) an approximation of feature sinuosity; 2) association with other features (e.g. does it cross over other feature types?); and 3) a measure of feature width and shape. The template creation program would then be used to automatically generate an appropriate set of rotated templates.

- *Compatibility with Multiple Data Input/Output Formats*. The AFT software operates using data captured on black and white scanners. System input capabilities are being expanded to include color scanners. Color adds a new dimension to the system. This extra dimension could augment the tracking logic of the AFT by providing only those *layers* within the data base that a particular feature can exist on.

- *Confusion Points Parameter File* . The ability for the cartographer to locate, *a priori*, any obstacles along the feature segment that might cause incomplete trackings for the tracking algorithm as a pre-vectorization step is being included. In places where the algorithm gets confused due to symbology conflicts, one or more additional points could be entered using the stream mode capability of most digitizers. When the tracking algorithm reaches a confusion point, it immediately accepts the point, and reinitiates the tracking afterwards.

Conclusion

The AFT system is capable of becoming an advanced automated cartographic data entry system utility. The Automatic Feature Tracking system is not the panacea to the entire digitization problem. Instead, the AFT system would serve to enhance the map conversion process by reducing the slow, error-prone, labor-intensive manual digitization processes. The application of automatic procedures, or even semi-automated procedures, can significantly shift the burden of these labor-intensive tasks from the operator to the computer. If a rapid input device like raster scanning could be followed with an automatic feature discrimination system on an edit/tag workstation, great improvements in production throughput rates for product generation can be realized. In most instances, AFT will not supplant existing digitization and edit/tag systems but, instead, will augment each to bring them to their potential.

ACKNOWLEDGEMENTS

The AFT system was developed at PGSC by a dedicated and talented team. Along with the author, Mr. Bradley Barnell, Mr. Andrew Douglas, Dr. Michael Gillotte, Jr., Mr. Steven Haehn, Ms. Marsha Rapke, Dr. Jon Sanders, and Mr. John Snyder comprised the AFT team. The author would like to express his gratitude to them for were it not for their devotion to cartography, the AFT system would not have been realized.

REFERENCES

Antell, R.E. (1983), "The Laser-Scan Fastrak Automatic Digitising System," Proceedings, Fifth International Symposium on Computer-Assisted Cartography, Auto Carto V, Crystal City, Virginia, August 22-28, pp. 51-64.

Boyle, Ray (1979), "Cartography in 1990," Proceedings, International Symposium on Cartography and Computing, Auto Carto IV, Reston, Virginia, November 4-8, pp. 40-47.

Boyle, A. Raymond (1982), "The Status of Graphic Data Input to Spatial Data Handling Systems", in Peuquet, Donna, and John O'Callaghan, eds. 1983. Proceedings, United States/Australia Workshop on Design and Implementation of Computer-Based Geographic Information Systems. Amherst, NY: IGU Commission on Geographical Data Sensing and Processing. pp. 13-20.

Callahan, George M., and Frederick R. Broome (1984), "The Joint Development of a National 1:100,000-Scale Digital Cartographic Data Base," Technical Papers of the 44th Annual Meeting of the ACSM, Washington, D.C., March 11-16, pp. 246-253.

Fegeas, Robin G. (1983), "Modularization of Digital Cartographic Data Capture," Proceedings, Fifth International Symposium on Computer-Assisted Cartography, Auto Carto V, Crystal City, Virginia, August 22-28, pp. 297-306.

Franklin, Dennis P. and Garry L. Holmes (1978). Overview of Automated Cartography Efforts at DMAAC. Proceedings, Sixth International Symposium on Automated Cartography, Auto Carto VI, Ottawa, Canada, October 16-21, pp. 438-447.

Penney, Robert A. (1979), "The Ascendancy of Digital Cartography in DMA's Future," Proceedings, International Symposium on Cartography and Computing, Auto Carto IV, Reston, Virginia, November 4-8, pp. 236-243.

Peuquet, Donna J. (1979), "Raster Processing: An Alternative Approach to Automated Cartographic Data Handling," The American Cartographer, vol. 6, no. 2, pp. 129-139.

Peuquet, Donna J. (1981), "An Examination of Techniques for Reformatting Digital Cartographic Data/Part 1: The Raster-To-Vector Process," Cartographica, vol. 18, no. 1, pp. 34-48.

Peuquet, Donna J. (1982), "Vector/Raster Options for Digital Cartographic Data", in Peuquet, Donna, and John O'Callaghan, eds. 1983. Proceedings, United States/Australia Workshop on Design and Implementation of Computer-Based Geographic Information Systems. Amherst, NY: IGU Commission on Geographical Data Sensing and Processing. pp. 29-35.

Selden, David D. (1986), "Automated Cartographic Data Editing: A Method for Testing and Evaluation," Proceedings, 46th Annual Meeting of the ACSM, Washington, D.C., March 16-21, pp. 5-14.

Starr, Lowell E. (1986). Special Report, Mark II: The Next Step in Digital Systems Development at the U.S. Geological Survey, The American Cartographer, vol. 13, no. 4, pp. 368.

THE INWARD SPIRAL METHOD: An Improved TIN Generation Technique and Data Structure for Land Planning Applications

David G. McKenna
Foundation Land Company
4176 Burns Road
Palm Beach Gardens, Florida 33410

ABSTRACT

This paper introduces the concept of the Triangulated Irregular Network (TIN) for computer representation of topographic surfaces. Discussion focuses on the TIN's benefits for interactive topographic modeling and site design applications. The paper then presents an alternative method of TIN generation termed the Inward Spiral Method. This method of TIN generation represents an improvement over previous methods by maintaining the integrity of site boundary edges and by automatically augmenting sparse or widely disparate data sets. The paper concludes with a discussion of the method's data structure and a sampling of its potential applications.

INTRODUCTION

The triangulated irregular network (TIN) is a topological data structure used to represent three-dimensional topographic surfaces. The TIN was developed out of the desire of geographers and cartographers for a more accurate and efficient means of collecting and storing topographic data in a digital format. A TIN may be visualized as a set of triangles which connect surface data points in a continuous coverage of irregularly-shaped triangular facets. (Figure 1)

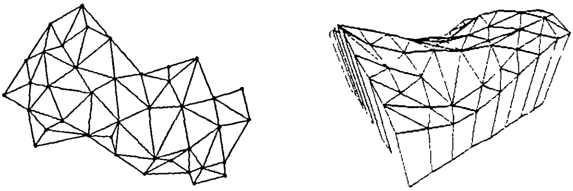


Figure 1 TIN representation of a topographic surface

TIN representations of topographic surfaces have several advantages over more commonly used grid representations. Mark (1975) and Peucker, et al (1976) have convincingly demonstrated that TIN systems result in a more accurate surface representation with far less storage, and McCullagh and Ross (1980) have shown that the generation of TIN surfaces can be accomplished much faster than the generation of gridded surfaces.

Grid systems do not permit vertical surfaces, and irregular boundaries or interior holes in a surface area are difficult to define with a grid. TIN systems, on the other hand, can describe nearly any surface, including those with holes, irregular boundaries, or vertical surfaces. In addition, the resolution of a gridded surface representation is limited to the resolution of the superimposed grid, while a TIN representation is limited only to the resolution of the original data.

THE TIN AS AN INTERACTIVE SITE DEVELOPMENT TOOL

These benefits have been exploited for a variety of applications. The TIN is now commonly utilized by automated survey systems, contour map generation software, earthwork calculation software and geographic information systems. Most of these applications employ the TIN purely as an internal data structure - that is, they utilize the TIN as a structure for storing and retrieving topographic data. But perhaps the most exciting advantage of the TIN, and until now the one most underutilized, is the TIN's tremendous potential as an interactive site design and development tool.

The irregular structure of the TIN is well suited to interactive design applications because it allows a surface to be freely manipulated and edited. Surface points, for example, can be moved in any direction without affecting the data structure of the original surface. Points can be added to or deleted from a TIN and the change accommodated by a simple, local triangulation of the altered triangles. (Figure 2)

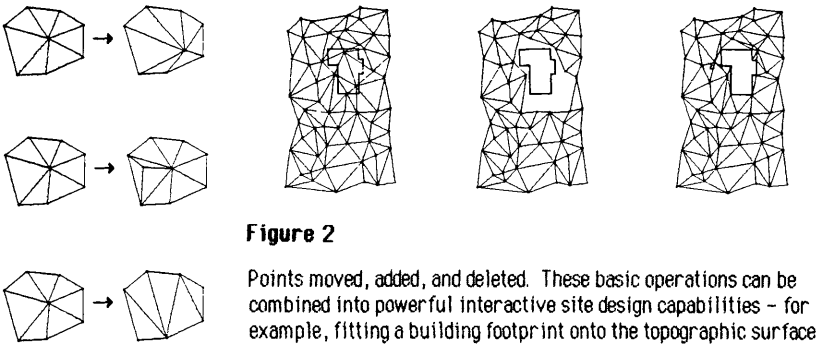


Figure 2

Points moved, added, and deleted. These basic operations can be combined into powerful interactive site design capabilities - for example, fitting a building footprint onto the topographic surface.

These advantages are being recognized by developers of three-dimensional site modeling systems. Several turnkey CAD vendors, for example, now offer the TIN as their topographic data structure. The use of the TIN in microcomputer-based CAD and engineering systems is also growing. The Inward Spiral Method of TIN generation described here is a component of one such microcomputer-based system.

METHODS OF TIN GENERATION

There are several methods of generating a triangulated irregular network from a set of data points. The process of each method is essentially a problem of "connect the dots;" that is, determining the connections between data points that will yield the best overall triangulation. The "best" triangulation is that which most accurately describes the surface being represented. In practice, this has proven to be the triangulation in which the triangles are most equilateral in shape.

Triangulation methods fall into two general categories. Methods of the first category actually involve two steps. The first step generates an initial, arbitrary network of triangles from the surface data points, while the second step refines the network by optimizing triangle shapes. Examples of this category can be found in the work of Gold, et al (1977) and Mirante, et al (1982).

The second general category of triangulation methods consists of methods which seek to generate the optimal triangulated network in a single step. These methods exploit the geometric principles underlying the organization of a TIN and produce what has come to be called the Delaunay tesselation (or triangulation) of a set of data points. Examples of these methods can be found in Brassel and Rief (1979), McLain (1976), McCullagh and Ross (1980) and Tarvydas (1983).

All these methods of TIN generation produce suitable triangulated networks but have several disadvantages. Some require that data be input manually, thus diminishing some of the economic benefits of automation. Some methods do not order data points or resultant triangles in an efficient and flexible manner, making interactive edit operations difficult. Some methods employ localized search procedures which, in extremely sparse or disparate data sets, can produce overlapping triangles. Most notably, all the reviewed methods typically encounter problems at concave boundary edges. Triangles may be generated outside the boundary, or triangle edges may intersect boundary lines. These problems can be eliminated only with substantial difficulty and loss of overall efficiency.

THE INWARD SPIRAL METHOD FOR TIN GENERATION

The Inward Spiral Method for generating triangulated irregular networks utilizes many concepts of previous methods. It introduces several enhancements to existing methods and is a superior method for certain applications. The method was developed to fit the requirements of SCHEMA, a three-dimensional modeling system being developed at the Harvard Laboratory for Computer Graphics and Spatial Analysis. One of the system's primary applications is the modeling of urban areas, and the special requirements involved with this application dictated that a different triangulation method be devised.

SCHEMA structures an urban model around the street pattern of the city. Blocks enclosed by streets define the surface areas to be triangulated, and the streets themselves constitute the boundaries of the triangulated areas. It was of paramount importance, therefore, to maintain the integrity of the street edges. No triangle could be generated outside the street boundary, and no triangle edge could intersect a street edge. This constraint led to the development of the Inward Spiral Method.

The Inward Spiral Method generates an optimal triangulated network in a single iteration, it maintains the integrity of boundary edges and it minimizes the possibility of overlapping triangles by automatically augmenting sparse or widely disparate data sets.

The heart of any TIN generation method is the algorithm which determines the points to connect to form an optimal triangle. The Inward Spiral Method uses the algorithm devised by McLain (1976). The McLain algorithm operates by assigning two points as endpoints of a triangle edge, examining neighboring data points and applying Euclidean geometry to determine the point which, when connected to the assigned edge, defines the optimal triangle.

The efficiency of this method depends on the efficiency with which it can determine the best point for the creation of a new triangle. Obviously, if every point on the surface area were examined for each new triangle, the efficiency would be considerably diminished. The Inward Spiral Method addresses this problem by superimposing a rectangular grid over the data set (Tarvydas, 1983). Data points are sorted into rows and columns within the grid. The search for the point defining the optimal triangle, then, is limited to those data points in adjacent grid cells.

After a complete data set has been input, the boundary edges of the surface are tested against the cell size of the superimposed grid. If edges are longer than the grid cell dimension, additional data points are inserted along the boundaries such that the distance between all boundary points is less than the grid cell size. This process eliminates the long, narrow triangles which typically arise at boundary edges and greatly diminshes the possibility of an error occurring at a boundary edge.

The data is then sorted into rows and columns within the superimposed grid. Each cell of this grid is then examined in turn. When a grid cell within the boundary of the surface area is found to be empty, a new data point is generated within that cell and its elevation determined by a distance-weighted averaging of nearby data points. This procedure ensures that every grid cell within the boundaries of the surface area will contain at least one point, thus minimizing the possibility of the generation of overlapping triangles and enhancing the aesthetic appearance of the surface.

The data are now ready for triangulation. The first boundary edge is chosen as a starting baseline, and the McLain algorithm examines data points to the inside of the boundary edge. The point selected to be the best point for a triangulation from that edge is tested to determine whether a triangle drawn to that point will intersect any boundary edge. If the test shows that an intersection will occur, the point is flagged and the next best point tested. If no intersection occurs, the triangle is drawn and added to a list of triangles in the data structure. The process is repeated for each consecutive boundary edge.

When the boundary triangulation is completed, the method looks to the triangles thus formed and establishes new triangles on the interior edges of previous ones. This procedure is repeated for each triangle, working from the lowest numbered triangle to the highest numbered triangle, until no new triangles can be created. By this sequence of triangulation, the method traverses the data points in a spiral pattern moving inward from the boundary edges. (Figure 3)

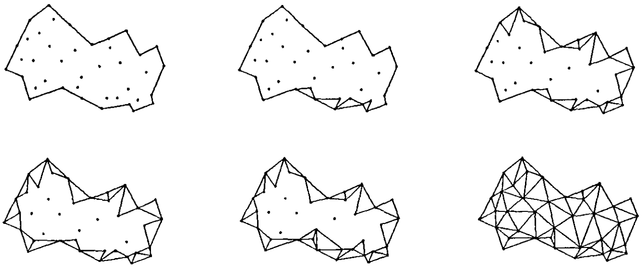


Figure 3 The Inward Spiral Method

The inward spiral pattern and the tests associated with the boundary triangulation ensure the integrity of the boundary with minimum effort, and thus the method represents an improvement over other methods in applications where maintenance of the boundary is essential. The automatic insertion of data points significantly improves the method's reliability and enhances the aesthetic quality of the resulting surface.

DATA STRUCTURES

There are two data structures that have been used in the generation of triangulated irregular networks. The first data structure regards the triangles themselves as the primary entities. Each triangle is an element in the data structure and is defined by pointers to the triangle's three vertices. The data structure also maintains pointers to each triangle's three adjacent triangles. (Figure 4) This is the more widely used of the two data structures and can be found in the work of McLain (1976), Gold, et al (1977), McCullagh and Ross (1980), Mirante, et al (1982), Tarvydas (1983) and others.

The second data structure regards the vertices of triangles as the primary entities. The network is defined by allowing each vertex to point to a list of the vertices connected to it by triangle edges. The list is sorted in clockwise or counter-clockwise order around the center vertex, starting at "north" in the local Cartesean coordinate system. (Figure 5) This data structure requires about one-half the storage of the first method. The structure is attributable to Peucker and Chrisman (1975) and can be found in the work of Fowler (1976).

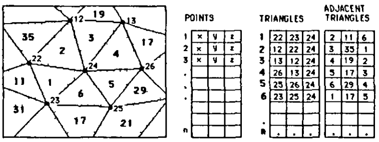


Figure 4 TRIANGLES data structure

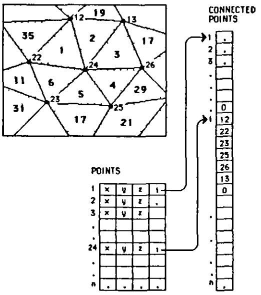


Figure 5 CONNECTED POINTS data structure

Each data structure has its own advantages and limitations, and each is suitable for different applications. A list of triangles, for example, is essential for efficient display functions such as hidden surface removal, surface shading, or fractual surface texturing. A list of connected points, on the other hand, is useful for a variety of editing operations such as "rubber-banding" of a modified surface, data point insertion and deletion or analytic functions, such as volume calculations or contour cutting. The Inward Spiral Method combines elements of each data structure into a dynamic, two-tiered data structure that is useful for all these applications. (Figure 6)

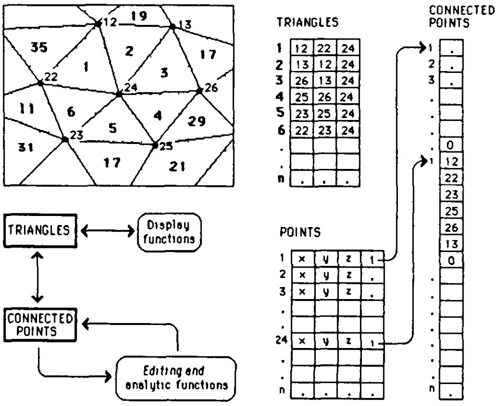


Figure 6 Inward Spiral Method data structure

An array (POINTS) contains the x, y and z coordinates of each data point. As triangles are generated in the triangulation process, the indices of the points defining each triangle are placed in the TRIANGLES list. This is similar to the first data structure described, but it is different in that the records of adjacent triangles are not kept. Instead, the list of triangles is used to generate the data structure of connected points (CONNECTED POINTS). A fourth element is added to the POINTS array which indicates the position in the CONNECTED POINTS list to which each data point refers. Editing and analytic operations are performed within the CONNECTED POINTS structure. If editing operations result in triangle changes, CONNECTED POINTS is transformed to a new, updated TRIANGLES list which is then used for display operations.

This combined data structure takes advantage of each of its component structures' best potential. It results in a system that is capable of both efficient interactive editing and sophisticated display. Although the data structure contains redundant information, it is of comparable size to the most commonly used data structure of triangles and adjacent triangles.

APPLICATIONS

One of the applications for which a triangulated irregular network is especially useful is interactive three-dimensional modeling. The irregular nature of a TIN allows it to be freely edited and manipulated, and the data structure of the Inward Spiral Method improves the efficiency of such operations.

One of the basic editing operations is the ability to move or "drag" a point along the topographic surface, with the edges connected to the point "rubber-banding" to the new point location. To accomplish this, a point is indicated with a cursor device. The search for the desired point involves only those points contained in the grid cell in which the cursor was activated and is, therefore, extremely fast. As the point is moved, the data structure of connected points instantly indicates the lines to be redrawn to create the rubber-banding effect.

Another basic operation involves adding new data points to an existing surface. This is done by determining the triangle which circumscribes the new point and connecting the point to the vertices of the circumscribing triangle. Mirante, et al (1982) describes a method for determining the triangle containing a point which employs extensive use of matrix algebra. The data structure of connected points, however, allows this function to be reduced to a few simple arithmetic operations (McKenna, 1985).

Data point deletion can also be performed very efficiently within the connected points data structure. The selected point is simply removed from the POINTS array and its associated points removed from CONNECTED POINTS.

The resulting polygon "hole" is then treated as the boundary of a surface with no interior data points. A boundary triangulation is performed and CONNECTED POINTS is updated to reflect the change.

The connected points structure is also useful for calculating the volume under the surface area. The volume under any single triangle is found by multiplying the surface area of the triangle by the average height of its three vertices. Computing the volume under an entire TIN in this fashion would require the computation of the area of each triangle and the average height of each triangle's three vertices. This would involve many separate area calculations, and since any vertex is shared by several triangles, the height of each vertex would be considered several times.

An alternative method of volume calculation takes advantage of the connected points structure. The number of points connected to a given point (i.e., the number of triangles which share that vertex) is easily determined. The volume under the entire surface, therefore, can be reduced to averaging the height of each data point by the number of points connected to it, summing the average height of all data points and multiplying the sum by the total area of the surface. This method represents a considerable savings, as each data point is considered only once, and no additional area calculations need be conducted.

A common application of the TIN is the automatic generation of contour maps. A contouring method which uses the connected points data structure is described by Peucker and Chrisman (1975). Other methods and discussions of this application can be found in most of the literature related to TINs.

TIN DISPLAY

Any computer-aided design system must be capable of fast, sophisticated three-dimensional display operations in order to be a truly useful design tool. The Inward Spiral Method employs the data structure of triangles for display operations. Triangles can be thought of as separate surface facets which combine to form the overall topographic surface. In this way, the TIN can be processed by many of the display operations commonly used in three-dimensional modeling systems such as hidden surface removal, surface shading or fracted surface texturing. Moreover, because a TIN is a continuous, connected surface, many of these operations can be simplified for TIN displays.

Hidden surface removal on raster display devices, for example, can be accomplished by a simple "back-to-front" display of the triangular facets. This process can be made more efficient by maintaining the values of the vectors normal to each triangle. Triangles whose normal vectors point away from the direction of view (i.e., those triangles which face away from the viewer) would not be seen from that viewpoint, and these triangles are never considered. The normal vectors can also be used for applying cosine shading to the surface (Figure 7) or for calculating slope or solar aspect of the surface.

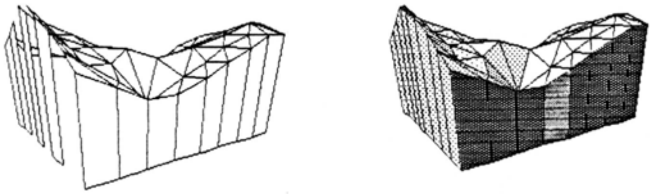


Figure 7 A TIN surface in perspective, with and without surface shading

CONCLUSION

The triangulated irregular network has been shown to be a superior system for topographic surface modeling. TIN systems execute faster than grid systems and produce more accurate surface representations with far less storage. In addition, TIN representations can be freely edited and manipulated and thus provide significantly greater potential for interactive surface modeling and site design.

The Inward Spiral Method for the generation of triangulated irregular networks is an improvement over other generation methods for certain applications. The method produces the most optimal triangulation in a single iteration. It maintains the integrity of boundary edges with minimum effort. And, by automatically augmenting sparse or widely disparate data sets, the Inward Spiral Method minimizes program error and enhances the aesthetic quality of the resulting triangulated network.

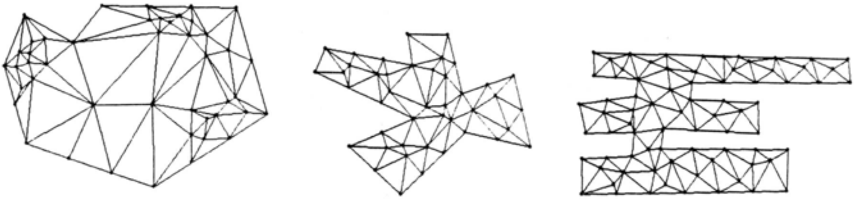


Figure 8 Complex surfaces successfully triangulated with the Inward Spiral Method

REFERENCES

Brassel, Kurt E., and Douglas Reif, 1979, "A Procedure to Generate Thiessen Polygons," Geographical Analysis, Vol. 11, No. 3, pp. 289-303.

Fowler, Robert J., 1976, "Database Implementation for the TIN Data Structure," Technical Report No. 11, "Geographic Data Structures" Project, U.S. Office of Naval Research.

Gold, C. M., T.D. Charters and J. Ramsden, 1977, "Automated Contour Mapping Using Triangular Element Data Structures and an Interpolant Over Each Irregular Triangular Domain," Computer Graphics, Vol. 11, pp. 170-175.

Mark, D. M., 1975, "Computer Analysis of Topography: A Comparison of Terrain Storage Methods," Geografiska Annaler, Vol. 57, pp. 179-188.

McCullagh, Michael J. and Charles G. Ross, 1980, "Delaunay Triangulation of a Random Data Set for Isarithmic Mapping," Cartographic Journal, Vol. 17, No. 2, pp. 93-99.

McKenna, David G., 1985, "The Triangulated Irregular Network," unpublished research paper, Harvard University Graduate School of Design, Department of Landscape Architecture.

McLain, D. H., 1976, "Two dimensional Interpolation From Random Data," The Computer Journal, Vol. 19, No. 2, pp. 178-181.

Mirante, Anthony, and Nicholas Weingarten, 1984, "The Radial Sweep Algorithm for Constructing Triangulated Irregular Networks," IEEE Computer Graphics and Applications, Vol. 4, No. 11, pp. 11-21.

Peucker, Thomas K., and Nicholas Chrisman, 1975, "Cartographic Data Structures," The American Cartographer, Vol. 2, No. 1, pp. 55-69.

Peucker, Thomas K., et al, 1976, "Digital Representation of Three-Dimensional Surfaces by Triangulated Irregular Networks (TIN)," Technical Report No. 10, U.S. Office of Naval Research, Geography Programs.

Tarvydas, A., 1983, "Terrain Approximation by Triangular Facets," Proceedings, Auto-Carto VI.

SCALE-BASED SIMULATION OF TOPOGRAPHY

Keith C. Clarke
Hunter College
695 Park Avenue
New York, NY 10021

ABSTRACT

A model of natural terrain is proposed, which describes topography as the summation of scale-dependent periodic spatial structure, and scale-independent fractional noise. These two elements can be varied, and appear related to each other by the sampling theorem and the scale at which the model is applied. The scale-dependent component of the model can be calibrated for any given landscape using Fourier analysis, as can the scale-independent component by measurement of the fractal dimension of a specific piece of terrain. The model is invertible, allowing simulation of terrain with specific surface characteristics. This paper develops the model mathematically, gives a description of the advantages and disadvantages of the model, shows examples of the use of the model for terrain simulation using a test data set, and discusses the cartographic uses of the model in automated mapping systems.

INTRODUCTION

The representation of terrain has been subjected to cartographic research for many years, and this research has been intensified by the introduction of automated cartographic methods. Traditional methods, such as representation by contours, hypsometric tints, and hatchures (Imhof, 1982), have been supplemented by stereo-pairs, hill shading, anaglyphs, wire grid perspectives; and, more recently, natural perspective views (Schachter, 1980). Describing or analyzing terrain, however, has been left as the domain of the geomorphologist, except for a few cartographic 'rules of thumb'. Cartographers enforce terrain rules such as the law of V's, the avoidance of enclosed depressions, and the imposition of the fact that water tends to flow downhill.

It is the intention here to propose that the cartographer's domain of interest extends further than the mere measurement and representation of terrain, indeed extends as far as mathematically describing terrain in such a way that the description can be calibrated to characterize different types of natural topography, and can be inverted to simulate terrain.

There are direct benefits to cartography with this approach, such as the ability to synthesize particular types of terrain to test various representation methods or map interpretation skills, or to add synthetic detail to thematic maps (Morrison, 1986). In addition, the approach offers possible research potential in the areas of optimal sampling, optimal map scale selection, and a method to create 'initial condition' landscapes for further modeling and simulations. The production of simulated depictions of landscapes also has potential uses as a cartographic context for computer graphics displays such as flight simulators and games (Hamilton, 1986).

The mathematical model of terrain advanced here has its origins in work in geology and geography and the structural analysis of terrain, and borrows from two bodies of theory. The first of these is that of spectral analysis, in which the 'model' of terrain is geometric, and consists of sets of trigonometric functions. The second is the body of work following the application of the concepts of Benoit Mandelbrot in geology and cartography. In this work, the model of terrain is based on the concept of self-similarity, and incorporates the idea of the fractional dimension (Mandelbrot, 1977).

The link between these two bodies of theory lies in their treatment of scale. Scale has always been of primary concern in cartography, and cartographers are familiar with the problems of generalization and loss of resolution associated with changing map scale. Similarly, scale is of critical concern in remote sensing and photogrammetry, and is here directly related to the problems of automated feature detection and recognition. The proposed model treats scale as a two step problem. Some of the generalizations made in cartography at smaller scales are designed to eliminate from the level of cartographic detail elements of the landscape of a particular size. Examples are the generalization of inlets and the elimination of islands along coastlines. The implication here is that some aspects of the landscape, for the purposes of description I will call them 'forms', have a particular map scale at which they can be depicted as significant. The range of scales at which the forms can be depicted are given by the sampling theorem, since the map should be large enough to depict at least one whole 'form' at one extreme, and should have a resolution such that at least two observations or spatial measurements fall along the length of the form at the other extreme.

The scales at which 'forms' exist in the landscape can be quantified using Fourier analysis. Beyond the limits of the sampling theorem, we can say little about the form since it cannot be measured, except that forms close to but beyond these ranges will produce non-random variation in spatial measurements. In particular, the too-large forms will introduce 'regional trends' into the data, while too-small forms will give the impression of local roughness of terrain or terrain texture. A convenient way to model this type of variation is to use the concept of self-similarity. Self-similarity implies that the spatial structures inherent within a landscape repeat themselves at all scales, i.e. a part of the whole resembles the whole. Self-similarity has not been found in many actual landscapes, and in itself seems a poor model of terrain, at least on earth. This is because earth-forming and earth-moving processes have been at work on almost all landscapes, for varying amounts of time, and as a result the landscape bears the 'forms' or manifestations of the scales at which these processes operate or operated in the past. Where these processes are largely absent, such as on Mars, self-similarity seems to hold (Woronow, 1981).

To summarize, the implication of this reasoning is that topography contains spatial forms with distinctive scales or 'sizes'. If these forms could be measured and extracted from terrain, the resulting variation would be the combined effect of forms beyond the measurement ability of a particular terrain map. This residual variation should show no scale-related structure and would be self-similar.

Several measures exist and have been used to describe form in a general sense, among them the autocorrelation function (Cliff and Ord, 1969) and the variogram (Woodcock and Strahler, 1983). Really these functions are expressions of the *neighborhood property*, in that they reflect how objects are related to each other as a function of the spacing between them. The proposed model of terrain uses Fourier analysis, and therefore suggests a trigonometric-periodic nature to the variation in elevation over space. The model could be described as being *scale-dependent*, since the separation of characteristic 'forms' by scale is the basis for the model.

When no scale-dependence exists, the neighborhood property is similar at all scales, implying that the phenomenon is self-similar, fractal, or *scale-independent*. Scale-independent phenomena are identical in structure (or lack of it) at all measurable scales but are not totally random, since they reflect structures at scales beyond those measurable. Since we can only deal with scales at which we can make measurements, we can usually only prove statistical self-similarity or scale-independence over a stated scale range.

Spatial form in a natural landscape is rarely distinct. Several forms at a variety of scales can exist together. Measurement of these forms must involve separation of the forms by scale and location. If these forms are extracted from a particular piece of terrain the residual could be considered purely random variation. Sources of this variation are measurement error, and variation due to stochastic processes; but, most significantly, variation exists because of scales beyond the limits set by the sampling theorem. The residual variance in terrain in this case is random at the scales over which it can be measured, so whatever scale is used for measurement the form will appear the same. Such a landscape has been termed self-similar (Mandelbrot, 1977) and can be simulated by fractional Brownian motion. It is not likely that fractal characteristics are displayed by terrain as a whole, a fact shown by Mark and Aronson (1984). If terrain itself was self-similar, the whole surface could be modeled by this single process (Goodchild, 1982).

Measurements describing this fractal type of form depend on the size of the measurement instrument, such that the relationship between measurements and the instrument size form a predictable ratio which can be used to determine the terrain's fractional dimension. The fractional, fractal or Hausdorff-Besicovich dimension is a real number, unlike the integers associated with dimensions in Cartesian geometry, and varies between 2 and 3 for a surface.

The natural landscape is a complex amalgam of spatial forms, each with its own associated scale, plus the scale-independent variation with fractal characteristics. It can be thought of as consisting of two elements, the scale-dependent part, reflecting the structures of objects at the various scales discernible in the landscape, and the fractal part, which is scale-independent. Measurements of spatial form therefore must be able to separate these elements, to split the scale-dependent part into its various scale objects and to quantify the fractal dimension of the residual variation. Simulation of terrain should involve inverting these measurements to produce a synthetic piece of terrain which is consistent with the derived measurements.

METHODOLOGY: MEASUREMENT AND MODELING OF SCALE-DEPENDENCE

Cartographic symbolization of three dimensional objects is based on the topographic surface, and its discrete representation, the Digital Elevation Model (DEM). The Geological Survey publishes these data, at scales of 1:250,000 and 1:24,000, with spatial resolutions of three arc seconds and 30 meters respectively. These data are integer elevations in meters, sampled at the nodes of a regular square grid. The DEM can be thought of as cartographic measurement of terrain at a particular resolution (the grid spacing), at a particular map size (the number of rows and columns in the DEM). DEMs have the properties of elevation, slope, volume, surface area, direction and angle of dip, texture, pattern and roughness; and they show the effects of scale-dependence and independence. Since topographic surfaces are assumed to be continuous, topology will be ignored, and all values on the surface are unique (no underground caves or overhangs).

Scale-dependencies are measured for calibration of the model using Fourier analysis of a DEM. This method is based on the concept that functions defining the surface of a DEM can be abstracted into sets of trigonometric series. Any surface, regardless of complexity, can be modeled by the sum of a set of sine and cosine waves with different wavelengths and amplitudes. This is indeed true if we can use an infinite set of trigonometric functions, but in reality we are constrained by the sampling theorem to wavelengths of a given range, the range from the fundamental to the Nyquist frequency. Most simple functions are well described by the sum of only a few sets of sine and cosine functions, and even some very complex functions can be adequately described using as few as ten sets (Tobler, 1975).

Fourier analysis (Rayner, 1971) computes a full set of trigonometric functions for the given range of wavelengths, and determines each one's contribution to the description of the curve under analysis. Those that make large contributions are valuable and can be used to invert the measurement process. Associated with each of these 'significant' trigonometric functions is a spatial 'wavelength'. This is the scale at which a periodicity exists. Fourier analysis can be used to reveal which scales are present in the data, and is useful in calibrating a scale-based model of terrain. Examples of the use of Fourier analysis to do this are provided by Davis (1973), Rayner (1972), and Bassett (1972).

Fourier analysis is directly invertible. This means that given the descriptors of the trigonometric components of the curve, the curve can be reconstructed with a level of accuracy dependent upon the number of descriptors used. If only the most significant descriptors are used in the reconstruction, the effect is to rebuild a regular spatial form, from which has been removed both variation beyond the ranges of the sampling theorem and that residual variation not attributable to the principal harmonics.

The same technique of Fourier analysis can be applied in one and two dimensions. Early work in Geography and Cartography on this problem was performed by Moellering and Rayner (1979), exclusively in one dimension, or as two one-dimensional series. Alternately, use of a two dimensional version of the functions allows the analysis of surfaces (Clarke, 1984; Rayner, 1972). The equation for a one dimensional series is:

$$z_t = \sum_{k=1}^{k=km} (A \sin k \omega x_t + B \cos k \omega x_t) \qquad [i]$$

where spatial resolution x is given by:

$$x = L / k \qquad [ii]$$

and L is the "length" of the map. The value omega is termed the angular frequency, or in spatial terms the number of cycles that a given periodicity goes through in a given distance, and is given by:

$$\omega = \frac{2\pi}{L} \qquad [iii]$$

The value km is the limit of the Fourier series, and is reached when x approaches twice the resolution of the data. The more complex two dimensional series is given by:

$$z_{x,y} = \sum_{k=1}^{k=km} \sum_{j=1}^{j=jm} (A_{kj} C_k C_j + B_{kj} C_k S_j + C_{kj} S_k C_j + D_{kj} S_k S_j) \qquad [iv]$$

where the sine and cosine terms are given by:

$$C_k = \frac{\cos(2k\pi x)}{L_x} \qquad [v.1]$$

$$C_j = \frac{\cos(2j\pi y)}{L_y} \qquad [v.2]$$

$$S_k = \frac{\sin(2k\pi x)}{L_x} \qquad [v.3]$$

$$S_j = \frac{\sin(2j\pi y)}{L_y} \qquad [v.4]$$

and where L is the length of the series in x and y as given by the subscripts. The power spectrum for the two dimensional series is an array, P given by:

$$P_{k,j} = A^2_{k,j} + B^2_{k,j} + C^2_{k,j} + D^2_{k,j} \qquad [vi]$$

The decision has to be made as to what constitutes a 'significant' harmonic to be used in the reconstruction and considered a scale at which a form occurs. Peaks in the power spectrum are used to signal a significant harmonic. Generally, if terrain is reconstructed from only a few harmonics, the surface will be both smooth and simple. The more harmonics are included in the inversion, the more the inverted Fourier surface will resemble its original. A quantitative basis for the inclusion of harmonics is the proportion of each harmonic's contribution to the total power, a value similar to a proportional contribution to variance.

Fourier analysis is comparatively simple to perform, shows significant scales directly, is applicable to two and three dimensional data, and is invertible, allowing a 'form' surface to be rebuilt from the significant parts of the trigonometric series. Two computational forms of the Fourier transform are in current use. The discrete transform has the advantage of conceptual simplicity and applicability to all data ranges. The fast-Fourier transform (Bloomfield, 1976), however, is vastly superior to the discrete transform in computational efficiency, but suffers from the problem of being applicable to only specific dataset sizes.

METHODOLOGY: MEASUREMENT AND MODELING OF SCALE-INDEPENDENCE

Measurement of scale-independence implies measurement of the fractal dimension. Fournier et al. (1982) provided the necessary algorithms for simulating surfaces from the fractal dimension for surfaces, and Dutton (1981) used an algorithm for lines and polygons.

This implies that the inverse transform is feasible, if the fractal dimension has been empirically derived beforehand.

Several different but related methods exist for the computation of fractal dimensions. The first set of methods relate to the computation of the fractal dimension of lines (and polygons). The simplest method is the 'walking divider' method. This method simulates the analog process of opening a set of dividers to a spacing d, then walking the dividers along the line, taking a total of N steps. Actually, N should contain a fraction of a step at the end of the line to reach the actual end point. The measured length of the line (L) is then N times the distance d. If the process is repeated, each time doubling the spacing of the dividers d, a set of pairs of observations of L and d can be derived. The natural logarithm of each of these values is taken, and a least squares linear regression is used to estimate the slope of the linear relationship between them. This slope (b), is then used to compute the fractal dimension f, which is simply 1 - b. Since the value of b is negative (i.e. the length of the line decreases as the spacing of the dividers increases) f is greater than 1 and is constrained to be less than 2.

A computer implementation of this algorithm has been performed by the author, and other versions exist (Shelberg et al., 1982). Goodchild (1982) has used a cell counting method, in which the number of joins between cells above and below a certain elevation is used as a proxy for line length. The same author also used a simple shape measure (perimeter length divided by area) for the areas within contours to obtain similar fractal dimensions to the walking dividers method.

Computation of fractal dimensions for surfaces has been performed in a variety of ways. The simplest method is to take advantage of the fact that any horizontal cross-section of a fractal surface should form a polygon with a fractal dimension which is the fractal dimension of the surface minus one. Since contours provide convenient cross-sectional polygons, these have been most frequently used (Goodchild, 1982; Shelberg and Moellering, 1983), while Burrough (1981) has used vertical cross-sections. Polygons within contours at sea level reflect coastal landforms, while those at high elevations reflect fluvial, mass wasting and glacial landforms. Scale-independence, statistical or otherwise, seems unlikely over these ranges. Shelberg and Moellering (1983) attempted to rectify this problem by interactive selection of the chosen isoline.

Mark and Aronson (1984) used the variogram as a mechanism for the computation of the fractal dimension. To compute the variogram for a surface, Mark and Aronson used sets of randomly-located points within the largest circle contained within the map area. About 32,000 such points were used for each surface. From the elevations of these points, the variance was computed as a function of distance. This relationship was used in a log-log regression to estimate the slope (beta) and thus the fractal dimension f as:

$$f = 3 - (\beta / 2) \qquad \text{[vii]}$$

Mark and Aronson computed fractal dimensions for seventeen digital elevations models with 30 meter resolution for a variety of landscapes. Of these, only one surface showed self-similarity at all scales measured, this being the Shadow Mountain 7 1/2 minute quadrangle in the Colorado Rocky Mountains. It may be possible that computations on this DEM involved data errors (Mark, personal comm., 12-14-86). All others to some extent, and some very strongly, showed periodicities due to spatial forms at particular scales. These results strongly suggest the validity of the proposed model.

A similar method was suggested by Burrough (1981). Burrough used the variogram, the Fourier power spectrum, the covariance, and the variance to compute fractal dimensions of vertical cross sections for a variety of different environmental phenomena. The use of the Fourier power spectrum was suggested by Fournier et al. (1982) as one of three possible approaches to fractal modeling and was used by R. Voss for the illustrations in Mandelbrot (1977). The method has been extended for use in image segmentation for image processing by Pentland (1984). The actual method consists of first computing the Fourier power spectrum, and then taking the slope (beta) of the log-log relationship between the power series and distance. In this case, the fractal dimension is given by:

$$f = 3 - (\beta / 2) \qquad \text{[viii]}$$

The similarities between this and the variogram approach are more than coincidence, since the variogram is the algebraic equivalent of the power spectrum. Burrough made estimates of fractal dimensions from observed data on variograms, block variances, covariance and the power spectrum, and found many different environmental data to show scale-independence. Burrough concluded "although some environmental data do appear to display the fractal property of statistical self-similarity at all scales, there are also many that show self-similarity over a limited range of scales, or over a few widely separated scales".

Fourier analysis allows the computation of the two-dimensional power spectrum for terrain. At every point in the power spectrum array, the one-dimensional distance of the harmonic combination can be computed, since for harmonic pair kx in the x direction and ky in the y direction:

$$d_{kx,ky}=[(L_x \;/\; kx)^2 + (L_y \;/\; ky)^2]^{(1/2)} \qquad [\text{ix}]$$

The fractal dimension can then be computed from a log-log fit of power and distance, taken across all values of k from 1 to the maximum, which is given by twice the resolution of the data.

$$\frac{d\ (\log P)}{d\ (\log d\ )} = \beta \qquad [\text{x}]$$

Beta can then be substituted into equation viii.

Another method for computing the fractal dimension of surfaces has been developed by the author (Clarke, 1986), and has been called the Triangular Prism Surface Area method. This technique is a three dimensional equivalent of the walking dividers method, and uses the surface area at a range of resolutions for a DEM to estimate the fractal dimension. In this case, the log-log relationship used to estimate the fractal dimension is that between the total area of the DEM surface, and the size of the cells into which it can be divided equally by powers of two.

The attractions of the Fourier power spectrum for the computation of the fractal dimension within the context of this work are multiple. First, the power spectrum is computed for a surface as part of the means of extracting significant periodicities in the scale-dependent part of the analysis. Secondly, this measure may be the best at estimating the fractal dimension since a larger number of scales are used in the log-log regression. Thirdly, and most importantly, the Fourier coefficients can be made fractal, and then the transform can be inverted to provide a simulated surface with fractal characteristics.

THE MODEL

The model of terrain postulated here may be expressed as follows. Elevation (z) within a DEM is given by:

$$z_{x,y} = T_{x,y} + cF_{x,y} + H_{x,y} \qquad [\text{xi}]$$

where:

$$T_{x,y}=\beta_0+\beta_1 x +\beta_2 y \qquad [\text{xii}]$$

defines a linear trend (very low frequency harmonic), and

$$F_{x,y}= \sum_{k=1}^{k=km} \sum_{j=1}^{j=jm} (A_{kj}C_kC_j+B_{kj}C_kS_j+C_{kj}S_kC_j+D_{kj}S_kS_j) \qquad [\text{xiii}]$$

The Fourier coefficients are provided in one of two ways. For significant harmonic pairs, the coefficients are derived empirically from real DEMs by Fourier analysis. During the inversion, insignificant harmonics are given zero Fourier coefficients, resulting in an inverse transform DEM showing scale-dependency only, H. In the case of scale-independency, the Fourier coefficients are given by:

$$A = B = C = D = \sqrt{}[P_{k,j}/4r^2] \qquad [\text{xiv}]$$

where r is 1 if k and j are zero, 2 if one but not both are zero, and 4 otherwise. The power is given by:

$$\log P_{k,j} = \alpha + \beta \log d_{k,j} \qquad [xv]$$

The values of A, B, C, and D in equation xiii can be either positive or negative, and if desired could be assigned random values such that their sum fell within limits. In the implementation of the model here, they were assigned values which were identical to those in the empirical Fourier analysis.

Finally, the relative contribution of the scale-independent component was controlled by scaling the range by a constant c. This was done because the linear regression provided a new set of Fourier coefficients for the full range of harmonics, including significant harmonics and those with very high frequencies. Significant harmonics often fall among the longer wavelengths, and can significantly influence the trend of the regression. Rather than filtering the data after the inversion to remove the high frequencies, instead the harmonic component was scaled by a factor designed to make the new surface mean and variance similar to the original. Thus the inverse transform involved two separate inverse Fourier transforms for the scale-dependent and the scale-independent components.

TEST APPLICATION

The model described above was tested using data from the Bell and Volfe Canyon areas in California's San Gabriel Mountains. This dataset had a fractal dimension of 2.193 computed by the Triangular Prism Surface Area method. Nine significant harmonics were found by the Fourier analysis for the area to have percentage contributions to the variance over 2.5 percent, with spatial wavelengths of 4 025 meters, 3 600 meters, 2 546 meters, 1 801 meters, and about 85 meters. These harmonics collectively contained 59.7% of the surface variance. Six of these harmonics made percentage contributions to the variance of over five percent.

A weak linear trend (r-squared of .38) was subtracted from the data which dipped to north north east. The remainder of the surface was subjected to Fourier analysis, and a log-log regression was performed between the raw power spectrum and spatial distance for each harmonic pair. The result was a weak linear relationship (r-squared of .32) with a slope of 1.162. Using equation viii, this yields a fractal dimension for the scale-independent part of the surface of 2.42, higher than the previous estimate. This regression was used to generate estimates of the Fourier coefficients in all harmonics, and the Fourier transform was inverted, scaled to reduce the variance to the same range as the original data, and added back to the linear trend. Finally, the inverted Fourier transform using the significant harmonics alone was added. Figure 1 shows a hill-shaded depiction of the Bell/Volfe Canyon data. Figure 2 shows the inverse Fourier transform using only the harmonics containing more that 2.5 percent of the surface variance, also hill-shaded. Figures 3 and 4 show the full application of the model. In the case of figure 3, a cutoff of 2.5 percent was used as a criterion for the inclusion of a harmonic as significant. In figure 4, a value of 5 percent was used.

DISCUSSION

The model proposed has various advantages and disadvantages for use as a terrain simulator. Advantages include the use of a single method, Fourier analysis, to incorporate two separate scale-dependent and scale-independent elements; the ease of inversion of the Fourier transform, the ability to base the model on parameters which can be measured from actual terrain, and finally, the fact that while a limited scale range is used here, the scale-independent component of the model makes the extension of the model across scales possible. The latter is important in computer graphics applications, where simulations are frequently required to zoom in and out on an area.

Disadvantages include the inefficiency of the current algorithm, which uses a discrete Fourier transform; the weak fit of the log-linear regression, which suggests that the residual may not be purely fractal; and the need to scale the inverted fractal surface back to the original surface variance range. Each of these shortcomings will be approached in future work.

In many respects, a terrain model can be judged by whether it produces terrain which 'looks right' in spite of the theoretical model from which the terrain was simulated. Nevertheless, the calibration of the model for different types of terrain in different parts of the world should produce statistics of geomorphological as well as cartographic interest.

The simulation of terrain may have applications in future automated mapping systems. Where thematic information is being depicted, for example, the precise type of terrain shown as background information need not be strictly accurate, especially at small scales. In these cases, an automated cartographic system could call up synthetic terrain with specific characteristics, rolling hills for example, and display them as relative topographic symbols. Alternatively, the system could provide synthetic landscapes for the development of context-free symbolization, which could then be transferred to a particular landscape or used to test the effectiveness of different representational methods. A distinct advantage of the scale-based model is that such systems could use the same model, regardless of the scale at which a particular map was to be produced.

Figure 1: Hillshaded Image of Bell Canyon DEM (120 x 120 pixels)

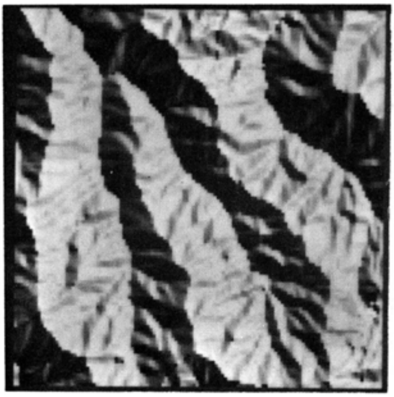


Figure 2: Bell Canyon Reduced to the Significant Harmonics

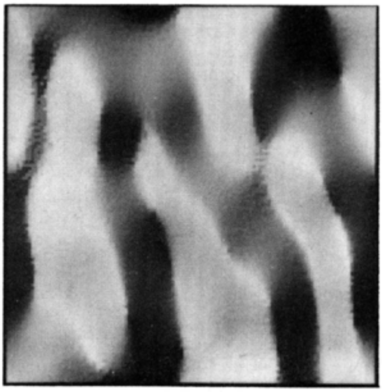


Figure 3: Simulated Terrain, Using Nine Harmonics, Fractal Dimension 2.42

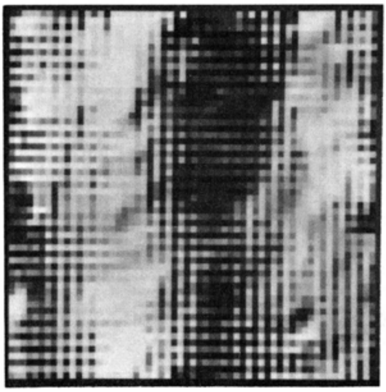
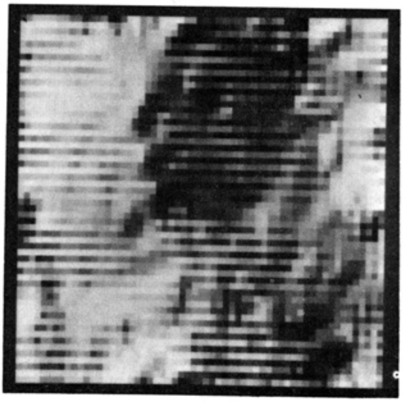


Figure 4: Simulated Terrain, Using Six Harmonics, Fractal Dimension 2.42



REFERENCES

Bassett, K. A. (1972) "Numerical methods for map analysis", *Progress in Geography*, 4, 219-254.

Bloomfield, P. (1976) *Fourier Analysis of Time Series: An Introduction*, J. Wiley, New York.

Burrough, P. A. (1981) "Fractal dimensions of landscapes and other environmental data", *Nature*, 294, 19 November, 240-242.

Clarke, K. C. (1984) "Two-dimensional Fourier interpolation for uniform area data", *Proceedings, ACSM-ASP Technical Meetings*, Washington D.C., 2, 835-845.

Clarke, K. C. (1986) "Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method", *Computers and Geosciences*, (in press).

Cliff, A. D. and Ord, J. K. (1969) "The problem of spatial autocorrelation", in Scott, A. J., *Studies in Regional Science*, Pion, London, 25-56.

Davis, J. C. (1973) *Statistics and Data Analysis in Geology*, J. Wiley, New York.

Dutton, G. H. (1981) "Fractal enhancement of cartographic line detail", *The American Cartographer*, 8, 1, 23-40.

Fournier, A., Fussell, D. and Carpenter, L. (1982) "Computer rendering of stochastic models", *Communications, ACM*, 25, 371-384.

Goodchild, M. F. (1982) "The fractional Brownian process as a terrain simulation model", *Proceedings, Modeling and Simulation Conference*, Pittsburgh, PA., 3, 1133-1137.

Hamilton, J. R. (1986) "A terrain modeling methodology for use in aircraft and surface-to-air missile encounter models", *Proceedings, Applied Geography Conferences*, 9, 228-233.

Imhof, E. (1982) *Cartographic Relief Presentation*, New York, DeGruyter.

Mandelbrot, B. B. (1977) *Fractals: Form, Chance and Dimension*, Freeman, San Francisco.

Mark, D. M., and Aronson, P. B. (1984) "Scale-dependent fractal dimensions of topographic surfaces: An empirical investigation, with applications in geomorphology and computer mapping", *Mathematical Geology*, 16, 7, 671-683.

Moellering, H. and Rayner, J. N. (1979) *Measurement of Shape in Geography and Cartography*, Dept. of Geography, Ohio State University, National Science Foundation Grant SOC77-113118.

Morrison, J. (1986) "Cartography: A milestone and its future", *Proceedings, Autocarto London*, London, Oct., 1986, 1-12.

Pentland, A. P. (1984) "Fractal-based description of natural scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, 6, 661-674.

Rayner, J. N. (1971) *Introduction to Spectral Analysis*, London, Pion.

Rayner, J. N. (1972) "The application of harmonic and spectral analysis to the study of terrain", in Chorley, R. J. (ed.) *Spatial Analysis in Geomorphology*, London, Methuen, 283-302.

Schachter, B. (1980) "Computer generation of shaded relief maps", in Freeman, H. and Pieroni, G. G., *Map Data Processing*, New York, Academic Press.

Shelberg, M. C., Moellering, H., and Lam, N. (1982) "Measuring the fractal dimensions of empirical cartographic curves", *Proceedings, AUTOCARTO V*, 481-490.

Shelberg, M. C. and Moellering, H. (1983) "IFAS: A program to measure fractal dimensions of curves and surfaces", *Proceedings, ACSM-ASP Technical Meeting*, Washington D.C., 483-492.

Tobler, W. R. (1975) "Linear operators applied to areal data", in Davis, J. C. and McCullagh, M. J., *Display and Analysis of Spatial Data*, London, J. Wiley.

Woodcock, C. E. and Strahler, A. H. (1983) "Characterizing spatial patterns in remotely sensed data", *Proceedings, Seventeenth International Symposium on Remote Sensing of Environment*, Ann Arbor, MI.

Woronow, A. (1981) "Morphometric consistency with the Hausdorff-Besicovich dimension", *Mathematical Geology*, 13, 3, 201-216.

AN ALGORITHM FOR LOCATING CANDIDATE LABELLING BOXES WITHIN A POLYGON*

Jan W. van Roessel **
TGS Technology, Inc.
EROS Data Center
Technique Development and Applications Branch
Sioux Falls, South Dakota 57198

ABSTRACT

Vector-based geographic information systems usually require annotation, such as a polygon number or attribute data, in a suitable location within a polygon. Traditional methods usually compute the polygon centroid, test the centroid for inclusion or exclusion, and select some alternative point when the centroid falls outside the polygon. Two problems are associated with this approach: (1) the text can be centered on the point, but may be placed in a visually awkward place, and (2) part of the text may fall outside the polygon and may overlap other polygon boundaries or other text labels. An algorithm is presented that circumvents both of these problems, by computing a number of horizontal candidate labelling rectangles (boxes) within a polygon from which a suitable selection can be made or from which one may conclude that the text label does not fit the polygon.

INTRODUCTION

The placement of feature annotation on maps is an important and difficult problem in automated cartography; it has been the subject of extensive research in recent years. In particular, the placement of labels in the vicinity of point features has been a subject of investigation as reported by Cromley (1986), Langran and Poiker (1986), and Mower (1986). However, two other types of problems also exist, namely, labelling of lines and polygons for which the applied methodology is still rather primitive in many systems. This paper will address the polygon labelling problem.

In a number of systems the polygon centroid (or a derivative) is used to locate a label within a polygon. As the centroid may fall outside the polygon, a fairly typical approach is to check this fact with a point-in-polygon method and then to shift the label to an arbitrary point inside the polygon. This method often leads to labels that appear in awkward locations and may partially overlap with the polygon boundary. The computed centroid is only a single point, whereas a label can best be represented by a minimum box bounding the text of the label. This minimum box must then be located such that (1) the box is wholly contained with the polygon, and (2) the box appears at a pleasing location. One may even want to repeat the text label at several locations when the polygon is large or complex.

The objective for this paper is to present a method for computing a number of candidate labelling boxes from which one or more suitable boxes may be selected. If no box can be found that is large enough to contain the label, one may then conclude that the label will not fit and some other placement action must be

* Publication authorized by the Director, U.S. Geological Survey.
** Work performed under U.S. Geological Survey contract 14-08-0001-20129.

taken.

In a subsequent process one may select a number of labelling boxes from among the candidate boxes, taking into consideration the characteristics of the polygon and the label. This process will only be briefly discussed.

APPROACH

The basic idea is to first divide the polygon into horizontal *strips*, where each strip boundary line passes through a vertex, and then to place vertical line segments on the polygon boundary segments located within the strips from which the boxes can be created by "sweeping" over these vertical segments in a left-to-right direction.

The polygon is first divided into strips by "drawing" a horizontal line through each of its vertices, as shown in figure 1. For a polygon with *N* vertices, this divides the plane into *N+1* strips, but the strips above and below the vertex points with maximum and minimum *y* coordinates are not of interest, so that *N-1* strips fully contain the polygon.

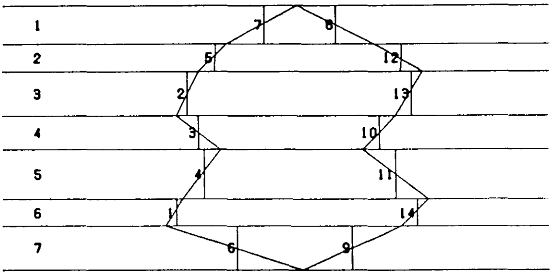


Figure 1.--A simple polygon with strips and vertical line segments.

The polygon line segments are subdivided by the strips such that each strip has an even number of divided segments located within the strips. The subdivided line segments will be referred to as *strip segments.* There are two strip segments in a strip for simple (no islands) convex polygons, and a multiple of two for non-convex polygons or polygons with islands (complex polygons).

Within a strip, a vertical line called a *vertical segment* can be placed through each strip segment to guide the formation of the candidate boxes (see figure 1). A vertical segment can be placed anywhere between the vertex points of a strip segment, but the strip segment midpoint has been selected. Other choices are the innermost or outermost vertex of a strip segment. The objective is to compute a set of maximal boxes, such that each box cannot be expanded further in the horizontal and vertical directions without crossing the polygon boundary. The present method only approaches this condition because each vertical segment becomes a part of a box, which, because of the midpoint location of the segment, will be partially outside the polygon boundary. An alternative is to locate the vertical segments at the innermost vertex of the strip segment; this guarantees that each box will be wholly inside the polygon, but some boxes will have zero area, so the space within the polygon will not be used as efficiently. Experience has shown that with the density at which natural resource applications polygons are usually digitized, the midpoint choice produces boxes that may have little "corners" outside the polygon. But in most cases, because the text box is usually smaller than the candidate labelling box, this is never graphically revealed.

The left and right boundaries of a box are formed by lines coincident with a left and right vertical segment, hereafter referred to as the left defining segment (*LDS*) and right defining segment (*RDS*). The bottom and top boundaries of the box are coincident with the respective lower and upper strip boundaries of the lowest and highest strip still contained within the box.

For instance, the box shown in figure 2 is constructed with left and right boundaries through vertical segments 5 and 10, and with the bottom and top boundaries coinciding with the lower and upper boundaries of strips 6 and 2. In general, a box is denoted as: *box(left, right, bot, top)*, where *left* and *right* are the vertical segment numbers of the *LDS* and *RDS* and *bot* and *top* are the bottom and top strip numbers.

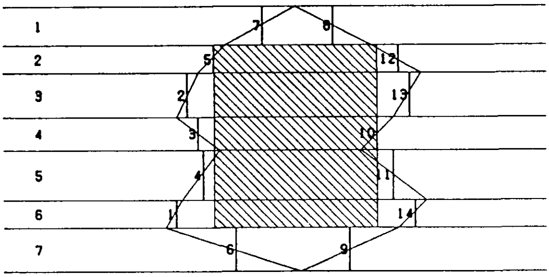


Figure 2.--Simple polygon with sample box.

After the polygon has been divided into strips and the vertical segments have been generated, the vertical sgments are sorted from left to right by increasing x coordinate. A vertical segment number is then assigned which is the sequence number in this sorted order.

Each vertical segment has an associated strip number $s(i)$, where i is the vertical segment number in sorted order, and $i = 1, \ldots, n$. The x coordinate of a vertical segment is denoted by $x(i)$. The vertical segments in a strip can be ordered from left to right as consecutive left and right pairs. In the left-to-right scan, only one pair is active in a strip at a time; a pair becomes inactive as soon as the right segment has been processed. Denote the left segment of this active pair by $l(s)$ and the right segment by $r(s)$. Then the following conditions hold for a maximal box: *boxmax(left, right, bot, top)*:

$$Pleft: \; x(left) \text{ is } max(x(l(s))), \; s=bot,\ldots,top$$

$$Pright: \; x(right) \text{ is } min(x(r(s))), \; s=bot,\ldots,top$$

From these two conditions, two resultant conditions for the top and bottom of the box can be derived, knowing that *Pleft* and *Pright* must be violated in the strips directly above and below the box, otherwise the box could be extended in these directions:

$$Ptop: \; x(l(top-1)) > x(l(top)) \textbf{ or } x(r(top-1)) < x(r(top))$$

$$Pbot: \; x(l(bot+1)) > x(l(bot)) \textbf{ or } x(r(bot+1)) < x(r(bot))$$

Another important condition is that the *LDS* and *RDS* of the box must coincide with the box boundary and therefore the following condition holds:

$$Pinc: \; bot \le s(left) \le top \textbf{ and } bot \le s(right) \le top$$

The approach taken for constructing a set of maximal boxes from the polygon is based on the above five conditions.

Each left vertical segment is potentially the *LDS* of one or more boxes. Condition *Pleft* states that the *LDS* has the largest *x* coordinate of all active left vertical segments in the strip range of the box; therefore only segments to the left of the *LDS* can influence the vertical range of the box, left vertical segments to the right have no bearing on it whatsoever. This suggests sorting the vertical segments by *x* coordinate and processing them in sorted order. Each segment can then be entered into a working array in which the boxes are created, and can be processed against other segments already entered to determine the range of the box for which it is the *LDS*.

The working array may consist of a number of rows and columns, each row corresponding to a strip in the polygon, and the columns representing left, right, bottom, and top elements for boxes that are being generated. The working array row corresponding to a strip will be referred to as the *strip entry*. Defining segments are entered into the working array at the strip entry of the strip in which they occur. Finished boxes in the working array may be inspected for selection immediately after they are generated or they may be saved in an output array for later processing.

Conditions *Ptop* and *Pbot* suggest how to determine the bottom and top limits of a box. The idea is to probe upwards and downwards until strips are encountered that violate *Pleft* and *Pright*. In the downward direction *Pleft* becomes invalid when *x(l(bot+1)) > x(l(bot))*. But since the violating segment has an *x* coordinate greater than that of the *LDS* it cannot as yet have been entered into the working array. And since the strips for a box are adjacent, it suffices to scan downward in the working array until an unused strip entry is encountered. The number of the strip before the empty entry then is recorded in the bottom column for the working array strip entry of the *LDS*. The same procedure is followed for the top. A set of adjacent active working array entries between two unused entries at the top and bottom will be referred to as a *cluster*.

When processing vertical segments by sorted *x* coordinate, *Pright* implies that each encountered right segment automatically becomes the *RDS* for one or more boxes. The problem is to determine the boxes that are terminated by the segment. Conditions *Ptop* and *Pbot* hold for all working array entries in the cluster because of the left segment conditions *x(l(top-1)) > x(l(top))* and *x(l(bot+1)) > x(l(bot))*. However, the *RDS* must also be on the box boundary. Therefore, subject to *Pinc*, all unfinished boxes in the cluster can be terminated by the *RDS* and be turned into completed boxes. Not all entries in the cluster may qualify.

The next problem is to consider which cluster entries that have produced boxes with the current *RDS* can live on in the left-to-right scan to produce more boxes, and how their top and bottom limits should be adjusted. Certainly the strip entry for the *RDS* must be terminated because the minimum right segment has been encountered. However, the other entries (satisfying *Pinc*) can continue if the top and bottom limits are adjusted. The strip entry for the *RDS* can therefore be reverted to the unused condition so that the cluster is either reduced by one strip (if the *RDS* strip is at the bottom or the top of the cluster) or the cluster is split into two parts. Considering the right segment conditions *x(r(top-1)) < x(r(top))* and *x(r(bot+1)) < x(r(bot))* of *Ptop* and *Pbot*, these can now be interpreted to mean that the strip of the *RDS* becomes *bot+1* as well as *top-1* for the other boxes in the new clusters (assuming without loss of generality that the *RDS* splits the cluster into two new clusters.) This means that the top and bottom columns for the entries in the working array for the new clusters must be adjusted to reflect these new limits.

The overall problem of generating candidate labelling boxes can be efficiently separated into two problems: (1) to compute the vertical segments, given the polygon definition, and (2) to compute the boxes from the vertical segments.

In the next section, the approach for computing the vertical segments will be discussed, but, because of space limitations, an algorithm will not be presented. Instead the emphasis of this paper will be on the precise algorithm for computing the boxes, which follows thereafter.

COMPUTING THE VERTICAL SEGMENTS

The approach used for computing the vertical segments has been to use a line-sweep approach (see, for instance, Sedgewick, 1983, chapter 24). The vertices are sorted on the *y* coordinate to divide the space into strips. Each *y* coordinate is then processed in turn, and its associated line segment number is either entered or deleted from an "active list" of line segments. With each *y*, a new strip is defined, of which the lower limit is set to the upper limit of the previous strip, and for which the new *y* becomes the upper limit. Line segments in the active band are then clipped against these limits and the *x* coordinates of the midpoints of the clipped segments are computed and entered into a list, together with the strip number in which they occur.

If the *y* coordinate following the current *y* coordinate is greater than the previous *y* coordinate in the polygon boundary, the line is removed from the active list. If this is not the case, the line segment is recorded in the list.

When the line sweep is completed, the list of strip numbers of the strip segment midpoints representing the vertical segments, is sorted by the *x* coordinate. This sorted list is then input to the algorithm discussed in the next section.

COMPUTING THE LABELLING BOXES

The algorithm for computing the boxes will be explained through a stepwise refinement process. The objective is to arrive at a fully developed procedure presented in Pascal. In the initial step there is only the procedure heading and ending, together with a characterization of as yet undefined code in the middle. The characterization will be presented as a Pascal comment in {} brackets. At each step this characterization will be further refined, but the entire set of derived code will not be repeated at each step; only the local expansion will be presented. The entire algorithm is then given at the end of the section.

The procedure is entered with a list of strip numbers for the vertical segments, which are in sorted order. This is the only input required for this algorithm, given that the output boxes are defined in terms of the *LDS* and *RDS*, and upper and lower strip numbers. Thus, the following code is first proposed:

Step 1

```
procedure boxes(strip: intarray; n:integer;
   var w: workarray; var box: boxarray);
var i, s, b, t, m, nr: integer;
begin
{produce the boxes}
end;
```

boxarray is a type of an array of records, each record with four fields: *left*, *right*, *bot*, and *top*. The working array is *w*, while *box* is the array that will contain the boxes on output. The working array is initially filled with zeros, and has a valid *w[0]* record. These records will be referred to as entries as before, where each entry has a *left*, *right*, *bot*, and *top* element (the record fields).

Although, for reasons of clarity, the procedure has an output array containing the finished boxes, one might instead inspect them on the fly. With this approach other information, such as the y coordinates of the upper and lower strip limits, and the *x* coordinates of the *LDS* and *RDS* must be imported to the procedure.

The process of producing the boxes is driven by the sorted vertical segments. For each segment the strip number of that segment, *s*, is an important variable. The working array entry corresponding to *s* will be referred to as the *current entry.* The following refinement is therefore made:

Step 2 {produce the boxes}

```
for i:= 1 to n do
   begin
   s:= strip[i];
   {process a vertical segment}
   end;
```

To process a vertical segment, two necessary actions must be taken. First, within the working array, the limits of the affected cluster must be established. Second, it must be decided whether the vertical segment is an *LDS* or an *RDS*, and appropriate action must be taken in each case. This leads to step 3:

Step 3 {process a vertical segment}

```
{establish cluster limits}
if w[s].left = 0 then
   begin
   {process LDS}
   end
else
   begin
   {process RDS}
   end;
end;
```

The *LDS* or *RDS* decision is made by checking the current entry. Since the *LDS* must come before the *RDS*, it suffices to check whether the left element of the current entry is empty (zero).

To understand how the cluster limits can best be determined, it is first necessary to know how the *LDS* is processed, which is shown in the following step:

Step 4 {process LDS}

```
w[s].left:= i; w[s].right:= 0; w[s].bot:= b; w[s].top:= t;
```

This step entails that the left element is set to the vertical strip number *i*, that the right side of the potential box is as yet unknown (*0*), and that the bottom and top are set equal to the cluster limits, *b* and *t*. It is necessary to assign *0* to the right element, even though *w* is initialized to *0*, because entries may be recycled in concave and complex polygons.

To find the cluster limits, the simplest idea is to scan the working array from the current entry in the upward and downward direction until empty entries are encountered. There is a better method, however, that makes use of information already entered in the cluster for previous segments. A cluster grows with new working array entries, which either begin a new cluster or are added to the top or bottom of an existing one . Consider the latter situation for the moment.

Four cases arise: new bottom and top for the current entry must be found, and for each it must be considered whether the current entry is at the top or the bottom of the existing cluster. If the new entry is at the the bottom, at position *s*, then the new bottom is known: *w[s].bot* = *w[s-1].bot+1* (strip numbers increase towards the bottom of the polygon). To find out whether the entry is at the bottom, one needs only to decrement the entry number, and see whether this entry is empty. If not, the current entry is at the top of the cluster.

If the current entry is at the bottom, *w[s-1].top* should contain the top of the cluster as established for some earlier state, not necessarily the previous state, depending on whether entries were made at the top or bottom of the cluster. In this case visiting *w[w[s-1].top].top* should yield a better estimate of the current top. However, this entry could point to itself, because the first entry for a cluster is necessarily confined to the strip for which it applies. Therefore, to make progress the next entry up (*w[w[s-1].top-1].top*) needs to be inspected. This progression is further pursued until the current top is reached.

This process can be illustrated with the following snapshot of the working array for one of the states related to figure 1

```
j   l r b t
-   - - - -
1 | 0 0 0 0                      where: j = working array
2 | 5 0 6 2  <-- top                        entry
3 | 2 0 3 3                              l = left
4 | 3 0 4 3  cluster                     r = right
5 | 4 0 6 3                              b = bottom
6 | 1 0 6 6  <-- bot                     t = top
7 | 0 0 0 0  <-- current entry
8 | 0 0 0 0
```

where vertical segment 6 must be entered, which lies in strip 7. As *w[7].left = 0*, it must be a left segment, and also since *w[8].left = 0* the entry occurs at the bottom of the cluster. Therefore *w[7].bot:= w[6].bot+1*, and the as yet incomplete working array entry 7 is set to: 6 0 7 0. To determine the top entry note that *w[6].top = 6*, and therefore points to itself. However, *w[w[6].top-1].top = 3*, and thus progress is made. But again *w[3].top = 3*, pointing to itself, so that *w[w[3].top-1].top = 2* needs to be inspected. This entry again points to itself, but decrementing by *1* points into an empty entry, so that the top of the cluster has been found. The complete entry for strip 7 becomes 6 0 7 2.

This leads to the following code for finding the top and bottom limits of the cluster for the current entry:

Step 5 {establish cluster limits}

```
b:=s+1;
while (w[b].left>0) do b:=w[b].bot+1;
b:=b-1; t:=s-1;
while (w[t].left>0) do t:=w[t].top-1;
t:=t+1;
```

It can easily be verified that the loops of step 5 will yield *b=s* and *t=s* when a new cluster is established in an empty part of the working array.

The remainder of the algorithm is dedicated to processing the *RDS*. On encountering a right segment, the following actions need to be performed:

Step 6 {process RDS}

```
for m:=t to b do
   begin
   if (w[m].bot<=s) and (s<=w[m].top) then
      begin
      {insert right limit}
      {output finished box}
      {update working array entry}
      end;
   end;
```

Each action must be performed for each of the working array entries within the cluster. Therefore, all actions are nested within a loop going from the top to the bottom of the cluster as established in the previous step. The test directly following the loop header enforces the right segment condition of *Pinc*. A design consideration at this point is whether the test may be better combined with the do loop in a "while do" loop. This option was not chosen considering the fact that the bottom entries in the cluster monotically increase (monotonic meaning a positive or zero step increment) away from the beginning original starting entry of the cluster in both the bottom and the top direction, while similarly the top entries decrease. Therefore, there may be more than one window where *Pinc* holds. Coping with these window limits would seem much more complex than performing a single test within the loop.

Inserting the right limit for the boxes to be generated from the cluster is simply a matter of inserting the *RDS* number in the right slot of the working array:

Step 7 {insert right limit}

```
w[m].right:=i;
```

With this insertion, the box is complete and can be output in a form depending on further processing to be performed on the box. It may be inspected for size or some related criterion immediately, or it may be stored in an output array for later processing. For the purpose of this paper it will simply be stored in an output array:

Step 8 {output finished box}

```
nb:=nb+1; box[nb]:=w[m];
```

The final step for each working entry in the cluster is to update the status of each entry. For the current entry, corresponding to the strip number of the *RDS*, the box has been completed, and hence the entry can be recycled for the next box. The entry (*m* = *s*) is therefore marked empty by setting *w[m].left:=0.*

For working array entries above and below the current entry, conditions *Ptop* and *Pbot* must now be enforced, because the strip of the *RDS* now becomes the strip at *top-1* and *bot+1* for the right segment conditions of *Ptop* and *Pbot.* Since *top* and *bot* in both conditions equal the current strip number, for strips above *s (s>m) s = bot+1*, therefore *bot = s-1*, so that *w[m].bot* must be set to *s-1.* Similarly, for strips below *w[m].top* must be updated to *s+1.* In step 9, therefore different actions are taken, depending on whether the entry indicated by the loop index is above, below, or at the current entry:

Step 9 {update working array entry}

```
if s>m then
   begin w[m].top:=s+1; w[m].right:=0; end;
if s=m then w[m].left:=0;
if s<m then
   begin w[m].bot:=s-1; w[m].right:=0; end;
```

This completes the stepwise development. Some overall improvements for efficiency can be made. Note that the *RDS* can be entered directly into the output array so that the working array actually does not need a right element. This results in the following algorithm, where *workarray* only has *left*, *bot*, and *top*, but *outarray* has all four elements:

Boxes Algorithm

```
procedure boxes(strip: intarray; n:integer;
    var w: workarray; var box: outarray);
var i, s, b, t, m, nb: integer;
begin
for i:=1 to n do
   begin
   s:=strip[i]; b:=s+1;
   while(w[b].left>0) do b:=w[b].bot+1;
   b:=b-1; t:=s-1;
   while(w[t].left>0) do t:=w[t].top-1;
   t:=t+1;
   if w[s].left=0 then
      begin w[s].left:=i; w[s].bot:=b; w[s].top:=t;end
   else
      begin
      for m:=t to b do
         begin
         if (w[m].bot<=s) and (s<=w[m].top) then
            begin nb:=nb+1; box[nb].left:=w[m].left;
             box[nb].right:=i; box[nb].bot:=w[m].bot;
             box[nb].top:=w[m].top;
             if s>m then w[m].top:=s+1;
             if s=m then w[m].left:=0;
             if s<m then w[m].bot:=s-1;
            end;
         end;
      end;
   end;
end;
```

NUMBER OF BOXES GENERATED

For practical usage of the algorithm, it is important to have some notion of the number of boxes that are generated as a function of the number of polygon vertices. Each box must be inspected as to its suitability to contain a label, and therefore the total number of boxes must be reasonable.

Each box is generated from within the cluster loop, which is nested within the vertical line segment loop. The number of boxes generated thus depends on the number of vertical segments, as well as the extent of the cluster for each line segment. Both of these factors depend on the geometry of the polygon. For a convex polygon in which there are no duplicate *y* coordinates (each *y* generates a unique strip boundary), the number of strips is *N-1* and the number of vertical segments is *2N-2*. The left segments build up a cluster with *N-1* entries. The right segments generate boxes. With each *RDS* only as many boxes as the size of the cluster can be generated, but the number is restricted because of the *Pinc* test right after the beginning of the cluster loop, where the influence of this test is determined by the geometry of the polygon. With each *RDS*, the size of the cluster is decreased by one so that for the first right segment at most *N-1* boxes can be generated, for the second *N-2*, etc., establishing an upper limit for convex polygons without islands of $(N-1)(N-2)/2$, from which it may be concluded that for convex polygons the number of boxes is $O(N^2)$.

For concave and complex polygons, the strips may contain a total of $O(N^2)$ vertical segments (Preparata and Shamos, 1985). This upper limit is approached in polygons with $O(N)$ spikes, each of which would behave similarly to a convex polygon from which one may conclude that a total of $O(N^3)$ boxes may be generated.

However, the average-case behavior of the algorithm is of more practical interest than the above worst-case complexity. But average-case complexity is a function of the spatial distribution of the vertices and is therefore nearly intractable.

Instead, an analysis was performed on 341 reasonably complex soils polygons with a minimum number of 4, an average number of 77, and a maximum number of 513 vertices. A maximum number of 84,536 boxes for a single polygon was generated for a polygon with 429 vertices, yielding a ratio of *number of boxes*$/N^2$ of 0.46. A maximum ratio of 0.74 was obtained for a polygon with 42 vertices and a minimum ratio of 0.06 for a polygon with 44 vertices. The average ratio was 0.19.

A least-squares fit for the data in the test data set was obtained for the model $y = a(x^b)$ where *y* is the number of boxes and *x* the number of vertices, with a resultant correlation coefficient of 0.98 and estimates for a and b of 0.18 and 1.99, respectively. Scatter plots of the residuals did not reveal any remaining trends.

Figure 3 shows a portion of the test data, with labels placed with the algorithm, where the lines in the label respectively represent the polygon number, the number of vertices, the number of strips, the number of vertical line segments, and the number of boxes. Labels that did not fit were replaced with the polygon number, placed within a small rectangle.

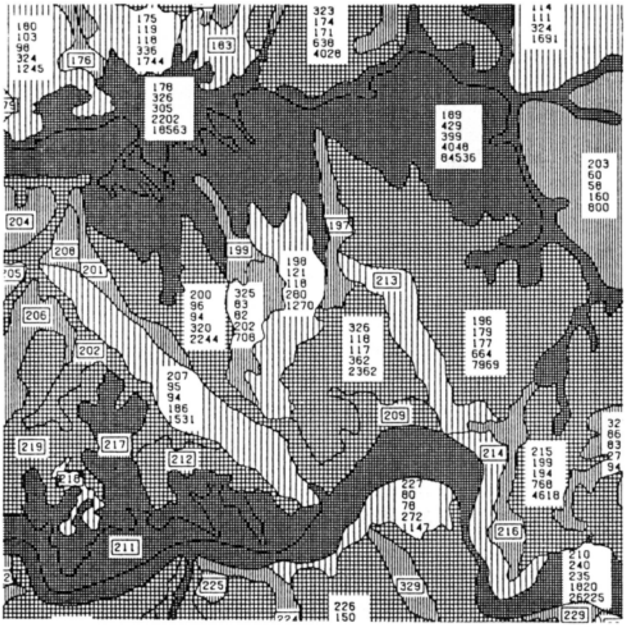


Figure 3.--Portion of test data set with label lines representing polygon number, number of vertices, number of strips, number of vertical line segments, and number of boxes generated.

One problem with the algorithm is the potentially large number of boxes that may be generated. Several solutions can be suggested. Unusually large polygons may be "weeded" to reduce the number of vertices. An alternative is to halt the box generation process when a sufficient number of boxes has been inspected. A third solution is to divide the polygon recursively into smaller polygons, which can be processsed using existing array sizes.

PERFORMANCE

The performance of the algorithm is closely related to the number of boxes generated. Time for the main loop is proportional to the number of vertical segments. For each left segment the cluster limits must be found. Because previous top and bottom information is used, not all cluster entries have to be inspected. Therefore, processing for a left segment is certainly $O(N^2)$ for convex and $O(N^3)$ for other types of polygons, because of the arguments used for the upper limits of the number of boxes. For each right segment the cluster limits are established, and then the cluster is scanned from top to bottom. An alternative arrangement would be to replace the top-to-bottom cluster loop with two loops scanning out from the current entry, while testing for the cluster limits. This would dispense with finding the cluster limits beforehand with the *while do* loops. This approach was tried for the test data set, and timings were performed, but no difference in performance could be detected. In both cases however, performance for the right segment would also be $O(N^2)$ and $O(N^3)$ for the two types of polygons, so that these limits also represent the worst-case complexity for the entire polygon.

Storage for the algorithm is the working array which must have as many entries as the number of strips (+1) so that basic storage is $O(N)$. Although the algorithm as presented stores the completed boxes in an output array, storing the completed boxes may not be practical, given the large number of boxes that may be generated. Therefore, since output storage is not a requirement because boxes may be inspected immediately, storage for the algorithm is *O(N)*.

SELECTING LABELLING BOXES

The algorithm may produce a large number of candidate labelling boxes from which only one or a few need to be selected to actually place the label. If only one box is to be considered, one might select the box with the maximum area. Each box, as it is generated, can then be inspected for area, and if the current box is larger than the previous largest box, it becomes the largest box, and so on. However, a much better visual result is obtained when the height-width ratio of the box somehow matches the height-width ratio of the minimum bounding rectangle of the block of text to be placed. Therefore, to select a better box, one may search for the maximum of a function of both total area and the aspect ratios of the label and the box. The following function has provided good results, and was used for figure 3:

$$f(a, rl, rb) = a\ e^{-|(rl-rb)|\ 0.4}$$

where *a* is the area of the box, *rl* is the height/width ratio of the label, and rb is the corresponding ratio for the box under consideration. This function modulates area according to the differences of the aspect ratios.

More complex strategies are in order for placing multiple labels. The number of labels might be determined based on the total area of the polygon, the area-perimeter ratio of the polygon as an index of the sinuosity of the polygon, the size and aspect of the label, etc. An additional problem with multiple labels is that the distance between labels probably should be optimized as balanced against the size of the selected boxes leading to the consideration of mathematical programming techniques.

REFERENCES

Cromley, Robert, G., 1986, A spatial allocation analysis of the point annotation problem: Proceedings of the Second International Symposium on Spatial Data Handling, International Geographical Union, p. 38-49.

Langran, Gail E. and Thomas K. Poiker, 1986, Integration of name placement and name selection: Proceedings of the Second International Symposium on Spatial Data Handling, International Geographical Union, p. 50-64.

Mower, James E., 1986, Name placement of point features through constraint propagation: Proceedings of the Second International Symposium on Spatial Data Handling, International Geographical Union, p. 65-73.

Preparata, P., and M. Shamos, 1985, Computational Geometry; Springer-Verlag, New York.

Sedgewick, Robert, 1983, Algorithms; Addison-Wesley, Reading, Mass.

PRACTICAL EXPERIENCE WITH A MAP LABEL PLACEMENT PROGRAM

Steven Zoraster
Stephen Bayer
ZYCOR, Inc.
220 Foremost
Austin, Texas 78745

ABSTRACT

Mathematical optimization algorithms have previously been shown to provide a solution to some map label placement problems. This article discusses computational and cartographic difficulties encountered and solved while implementing such techniques in commercial map label placement software. The solution of the optimization problem, the automation of label overplot detection, and the representation of deleted labels in the manner best suited to assist human intervention are discussed.

Despite remaining questions about the best way to implement this type of computer program, our work has shown that mathematical optimization techniques are an excellent method for performing map label placement. This conclusion is based on experience resolving labeling problems on maps compiled from typical data encountered in oil and gas exploration activities.

INTRODUCTION

Most algorithms designed for the placement of map labels around point symbols are characterized by sophisticated data structures and complex, rule-following logic (Hirsch 1982; Ahn and Freeman 1983; Ahn 1984; Basoglu 1984; Pfefferkorn, et al. 1985; Langran and Poiker 1986; and Mower 1986). In fact, these algorithms are attempts to solve a specific type of combinatorial optimization problem by heuristic or artifical intelligence procedures. Because artificial intelligence is not used to solve combinatorial optimization problems in non-cartographic applications, we have investigated alternate methods for solving the map label placement problem. Our work has produced a computer program which is suitable for production mapping. Our program solves the underlying optimization problem directly.

The optimization model for label placement was originally exploited by Cromley (1985), who developed an overplot resolution algorithm for point symbol labels based on a linear programming relaxation procedure and interactive detection of overplots. Its use was extended by Cromley (1986) and by Zoraster (1986), who have both used integer programming techniques along with automated detection of label overplots.

This paper provides a review of the optimization model for label placement, compares this type of algorithm to other placement algorithms, and discusses various implementation problems.

THE OVERPLOT RESOLUTION ALGORITHM

Simple label overplotting problems can be resolved without sophisticated algorithms. An advantage of using an optimization algorithm for this purpose is that it can resolve extremely complex problems efficiently. The only requirement is that there be a finite number of label placement options for each label. A prioritized set of options is shown in Figure 1 by numbers placed around a map symbol (X). Given such a placement priority for each label, the best label placement for a large number of labels can be found by use of integer programming.

3	1	2
5	X	4
8	6	7

Figure 1. Possible Label Position Options

Linear Programming and Integer Programming

Linear programming is a mathematical optimization technique used to minimize (or maximize) the value of a linear objective function subject to linear constraints on the values that can be assumed by problem variables (Chvatal 1983). Given an N element vector **c**, an M by N matrix **A**, and an M element vector **b**, the standard linear programming problem is to choose the N elements of the vector **x** to maximize

$$\mathbf{c}^T\mathbf{x},$$

subject to the constraints

$$\mathbf{A}\ \mathbf{x} \leq \mathbf{b}.$$

Integer programming problems are linear programming problems with the added constraint that the elements of **x** must be integer. 0-1 integer programming problems require that the elements of **x** be either 0 or 1. Our overplot resolution program uses 0-1 integer programming to resolve overplot problems.

Many linear programming algorithm implementations are available in the form of software subroutine libraries. Only a few of these implementations handle integer or 0-1 variables efficiently. This difficulty is discussed in more detail later in this paper.

The Overplot Problem Formulated As a 0-1 Integer Programming Problem

Assume that each of the K labels on a map can be placed in one of N_k possible positions, one of which corresponds to deletion from the map. In the mathematical formulation of the overplot resolution problem, each position for label k will correspond to a single variable $X_{i,k}$ (i = 1,2,...,N_k) which can take on the values of 0 or 1. Only one variable corresponding to each label is allowed to be 1. This restriction is enforced by the following constraints:

$$\sum_{i} X_{i,k} = 1 \qquad k=1,2,\ldots,K. \tag{1}$$

If an overplot is possible between the i-th label position for label k, and the j-th label position for label m, then the following constraint needs to be enforced:

$$X_{i,k} + X_{j,m} \leq 1. \tag{2}$$

Each position for each label will have a penalty $W_{i,k}$ associated with its use. Normally the optimal position will have a penalty of 0 and other positions will have positive penalties proportional to the difficulty caused by attempting to associate a well label in that position with its correct symbol. The penalty associated with deletion of a label from the map will be the largest penalty. Our goal is to attempt to minimize the total penalty represented by

$$\sum_{i}\sum_{k} W_{i,k} \quad X_{i,k}. \tag{3}$$

subject to the constraint sets (1) and (2).

ALGORITHM IMPLEMENTATION

Three major concerns in the implementation of any map design software are program speed, the quality of the output, and the ease with which the results can be edited by a cartographer. For this label overplot program, speed is affected primarily by the algorithm used to detect overplots and the algorithm used to solve the optimization problem. The quality of the output is primarily affected by how close the integer programming algorithm can approach an optimal solution to the integer programming problem. The handling of labels deleted from the input map is the most important factor affecting final map editing. Each of these concerns is addressed in the following paragraphs.

Overplot Detection

To detect overploting efficiently, label placement data must be partitioned or sorted. Cromley (1986) has suggested a referencing label positions to a Thiessen diagram based on the map symbols, but we have found that less sophisticated label locating schemes are cost effective for our application. We have used a simple sweep algorithm in which the labels, map symbols, and line segments are first sorted according to their X coordinates and then subject to a hierarchy of tests to detect actual overplots between items whose projections on the map X-axis overlap.

We have implemented a sequence of tests which allow most potential conflicts to be eliminated from consideration quickly. Intelligent programming strategies then make each required interference check simple. For example, when checking for the intersection between an arbitarily oriented label and a point symbol, the point symbol's position is rotated and normalized with respect to the orientation and center of the label, so that the interference check requires measuring only the magnitudes of each of the coordinates of the point symbol in the new coordinate system.

Practical Integer Programming Algorithms

Integer programming problems are generally difficult to solve. For large problems it is often impossible to obtain a feasible solution that is close to the optimal solution using general purpose software. Special techniques based on the structure of the constraints and on the structure of the objective function, along with careful problem formulation are required to solve such large problems (Johnson et al. 1985). A map label placement problem with hundreds or thousands of labels and hundreds of constraints creates a large integer programming problem. Most of the computer time required by our program is spent solving the integer programming problem.

Fortunately, the structure of the overplot resolution problem can be exploited to obtain efficient solutions. The optimization algorithm we use has been developed specifically to solve this map labeling problem. Our solution technique uses Lagrangian relaxation of the interference constraints which are included in the objective function subject to multiplication by a vector of Lagrangian weights. Using matrix notation, the problem we solve is

$$\text{min:} \quad \mathbf{w}^T\mathbf{x} - \mathbf{d}^T[\ \mathbf{1} - \mathbf{A}\mathbf{x}\ ]$$

$$\text{subject to:} \quad \mathbf{P}\mathbf{x} = \mathbf{1}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{1}$$

along with the 0-1 restriction on the elements of x and the restriction $\mathbf{d} \geq 0$. The equality constraints are simply the set (1) written in matrix format, and the inequality constraints are the pairwise conflict constraints (2), again written in matrix format. The elements of the vector **d** are the Lagrangian weights and are approximations to the dual variables of a linear programming relaxation of the original integer programming problem.

The Lagrangian weights are adjusted in an iterative manner according to the "subgradient optimization" method (Held, et al. 1974). When using this algorithm, the penalties for label variables corresponding to unsatisfied label interference constraints are increased at each iteration, while the penalties for label variables corresponding to constraints that are not binding are decreased. The amount of change is reduced periodically in a manner that guarantees algorithm convergence. On most maps this algorithm produces what appears to be very close to an optimal label placment. We have encountered no maps for which this algorithm does not produce at least a reasonable answer. Of course a map which is very densely labeled may require many label deletions to produce a solution.

Many other algorithms exist to solve integer programming problems. We have not seriously investigated the wide range of solution techniques available. Research to determine the best algorithm for this particular problem would be very useful.

Deleted Labels

The optimization algorithm performs the majority of work required to resolve label overplotting, but many maps will benefit from interactive editing to complete the label placement process. One type of data required during an editing session is information about exactly which labels have been deleted in order to obtain a solution. Two obvious graphical methods for recording this information are available. Deleted labels may be written to a distinct graphic layer of the output picture file or to a digital copy of the original input from which all other labels have been removed. Writing the labels to a distinct layer on the output picture is most useful if the program user has access to a color graphic device. Since many users of our software work on monochrome terminals, we have chosen to write the deleted labels, along with their associated symbols, to a second picture.

Better solutions to the problem of handling deleted labels might involve the automatic creation of insert maps at a larger map scale or the display of all deleted labels in a table outside the map border. The labels in the table could be referenced to their true positions on the map by a special class of symbols used to replace those symbols corresponding to deleted labels. Implementation of either of these options would require much work.

ADVANTAGES OF OPTIMIZATION APPROACH TO LABEL PLACEMENT

Most previous algorithms designed to solve the point label placement problem have not made use of mathematical optimization algorithms, but there is nothing in the reported problem formulations which precludes the use of this type of solution. Combinatorial algorithms described in the literature of mathematical optimization make as much use of mathematical techniques as is practical; these include algorithms designed for business planning, transporation analysis, personnel assignment, the allocation of military resources, and many other applications. In each of these domains heuristic techniques are used only when mathematical techniques fail because mathematical techniques produce better solutions. We believe that computer programs based primarily on optimization algorithms will out perform algorithms based on heuristics or artificial intelligence in this application as well.

There are at least two other significant reasons to favor the use of mathematical optimization algorithms to place map labels. First, mathematical optimization algorithms are easily implemented in procedural languages. Although there has been much talk about object oriented languages such as LISP or PROLOGUE for cartographic applications, most mapping systems are still coded in languages such as FORTRAN or C. Because cartographic problems often are mathematical in structure, procedural languages will continue to be used for the implementation of many cartographic programs. The program we have developed for map label placement is coded in FORTRAN 77.

Second, this type of algorithm has been proven using our computer program on many complex maps created by the oil and gas industry. To our knowledge, none of the other techniques previously presented in the cartographic literature as solutions to this problem have actually been used in production mapping. Figures 2 and 3 show label placements before and after overplot resolution for a small map with 40 point symbol labels. Similar quality results have been obtained on much larger data sets. For example, problems with over 2100 point symbol labels and over 1600 label conflicts have been executed on a VAX 750 in less than 1 CPU hour.

CONCLUSIONS

The optimization model for map label placement provides a practical solution to an important problem in automated cartography. Its usefulness has been proven in the compilation of maps with thousands of point feature labels and hundreds of label conflicts.

There are several unresolved technical problems in the implementation of this type of algorithm. Research to compare different ways of solving the optimization problem would be very useful. Also, better methods for displaying information about labels deleted from a map and methods to assist the interactive editing of the map created by this type of algorithm need to be investigated.

REFERENCES

Ahn, J. 1984. Automatic Map Name Placement System. Image Processing Laboratory Technical Report 063; Electrical, Computer, and Systems Engineer Department, Rensselaer Polytechnic Institute, Troy, New York.

Ahn, J. and Freeman, H. 1983. A Program for Automatic Name Placement. Proceedings, AUTO-CARTO VI, Vol. 2, pages 444-453.

Basoglu, U. 1984. A New Approach to Automated Name Placement Systems. Ph.D. Dissertation, Department of Geography, University of Wisconsin-Madison, available through University Microfilms International.

Chvatal, V. 1983. Linear Programming. W. H. Freeman and Company.

Cromley, R.C., 1985. An LP Relaxation Procedure for Annotating Point Features Using Interactive Graphics. Proceedings, AUTO- CARTO VII, pages 127-132.

Cromley, R.C. 1986. "A Spatial Allocation Analysis of the Point Annotation Problem," Proceedings, Second International Symposium on Spatial Data Handling, pages 38-49.

Held, M., Wolfe, P., and Crowder, H. 1974. Validation of Subgradient Optimization. Mathematical Programming, Vol 6, pages 62-88.

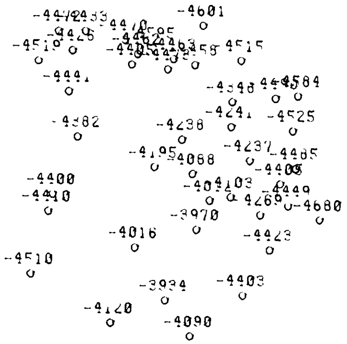


Figure 2. Point Symbol Labels With Overplotting

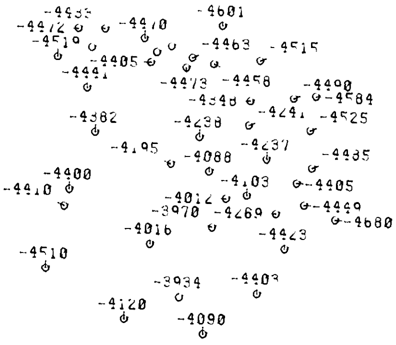


Figure 3. Point Symbol Labels With Overplots Resolved

Hirsch, S.A. 1982. An Algorithm for Automatic Name Placement Around Point Data. The American Cartographer, Vol. 9, No. 1, pages 5-17.

Johnson, E.L., Kostreva, M., and Suhl, V. 1985. Solving 0-1 Integer Programming Problems Arising from Large Scale Planning Models. Operations Research, Vol. 33, No. 4, pages 803-819.

Langran, M.S. and Poiker, T.K., 1986. "Integration of Name Selection and Name Placement," Proceedings, Second International Symposium on Spatial Data Handling, pages 50-64.

Mower, J.E., 1986. "Name Placement of Point Features Through Constraint Propagation," Proceedings, Second International Symposium on Spatial Data Handling, pages 65-73.

Pfefferkorn, C., Burr, D., Harrison, D., Heckman, B., Oresky, C., and Rothermel, J., 1985. ACES: A Cartographic Expert System. Proceedings, AUTO-CARTO VII, pages 399-407.

Zoraster, S. 1986. "Integer Programming Applied to the Map Label Placement Problem," Cartographica, Vol 22, No 3, pages 16-27.

AUTOMATIC RECOGNITION AND RESOLUTION OF SPATIAL CONFLICTS IN CARTOGRAPHIC SYMBOLISATION

William A. Mackaness
School of Geography, Kingston Polytechnic,
Penrhyn Road, Kingston Upon Thames, SURREY, KT1 2EE, U.K.

Peter F. Fisher,
Dept of Geography, Kent State University, Kent, Ohio 44242

ABSTRACT

One of the fundamental stages of the map design process is that of assigning and positioning representative symbols on a map. This is achieved in most computer cartography systems by user selection of the symbols, followed by the positioning of those symbols on the features nominated. If the symbols interfere with each other at all, the user must intervene. This user intervention is most undesirable because the changes made may have undesirable repercussions for the rest of the map; thus the process of change can become iterative and therefore time consuming. The objective of the research reported here is to automate this element of the design stage. Where points or lines overlap with each other or with others of the same type, three possible solutions may be recognised: re-symbolisation, re-location, and generalisation. The main area of research to date has been in the development and implementation of algorithms that apply cartographic license (clarification of information by localised small movements of features). The simplest problem occurs when points are not confined by any other map feature. This paper illustrates the progressive complexity of algorithms required as the solution becomes constrained by the increasing proximity of other features and argues the need for a solution that optimally clarifys such local conflicts whilst ´blending´ with the map as a whole.

INTRODUCTION

The Need For Automated Map Design

The techniques for transferring spatial data into computers (digitisation) and methods of accessing data (via geographical information systems- GIS) are becoming increasingly advanced (Green et al. 1985). This is in response to growing demands for efficient methods of handling the ever increasing volumes of spatial data. Parallelling this demand, A.I. techniques are being introduced at the data storage level, in the form of knowledged based GIS -KBGIS (Smith and Pazner 1984; Peuquet 1984). KBGIS have arisen directly from a real need to store and interrogate data efficiently, in a format that enables fast access to both raster and vector data. However these systems make no decisions on how the data is used and reveal limited information about the data.

The main component missing from the computer mapping environment is a system to control the design stage. This design role is normally performed by the cartographer (in consultation with the user), indeed ´nowhere in any production process do the needs of the users influence the nature of the product more than in the design stage´ (Page and Wilson. 1978, p157). But with increased ease of access, the design stage is being carried out by scientists and general users with little or no cartographic skill. With easy manipulation and selection of symbols, unwittingly patterns in the data are often either enhanced or suppressed, enabling the creation of ´cartographic monstrosities with unprecedented ease´ (Monmonier 1984, p389).

Potential Solutions

It is argued that the ideal solution to this problem is a computer system that mimics the role of the human cartographic expert. The advantage is that it would enable researchers with no cartographic skills to display field data in a variety of ways using optimal spatial designs (Mackaness et al. 1985).

In order for a system to mimic a human, ideally it must have equivalent human attributes: these include cartographic knowledge, a method of articulating that knowledge, and an ability to reason. Such mimicry is possible using artificial intelligence (A.I) techniques to construct an expert system. Expert Systems have been extensively reviewed in specialist literature (for example, Barr and Feigenbaum, 1982) and have been shown to be of interesting potential in design (R1), diagnosis (MYCIN), and prognosis (PROSPECTOR).

Present research is concerned with developing a system for one aspect of computer-aided cartography - namely evaluating and resolving spatial conflicts in map design. As discussed below, the problems are sufficiently complex that heuristic knowledge based methods may be the only way of resolving some of the central problems in this process.

PROBLEMS IN AUTOMATING MAP DESIGN

The process of cartographic design is a complex, interactive process between the user and the cartographer (see Morrison, 1980). It will depend amongst other things, on the requirements and knowledge of the user, and the facilities and experience of the cartographer. The complex process of map design can be considered under five headings.

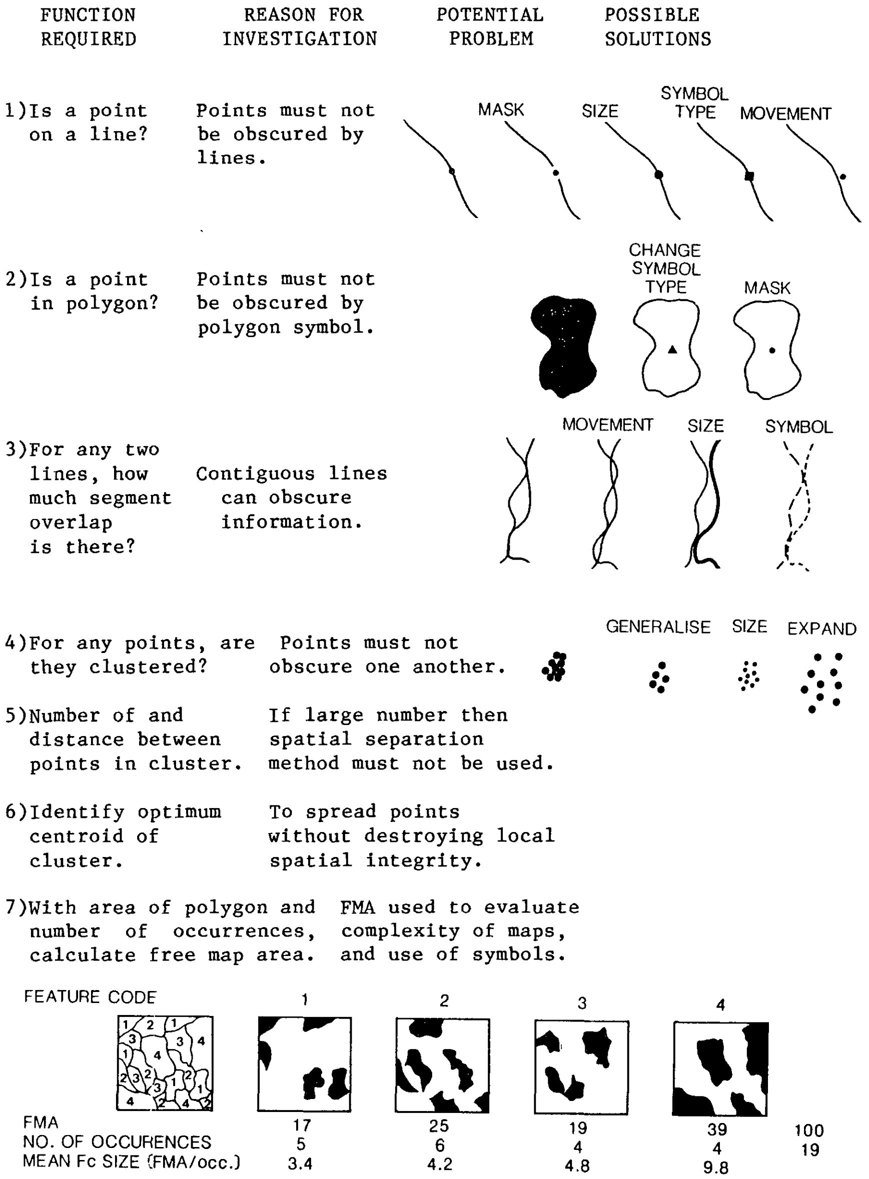
1. Gather from the user, information such as data to be displayed, map type, and map use.

2. To make decisions on levels of generalisation such as the acceptable levels of visual clutter and which base data to include.

3. Symbols can be assigned depending on the data categories to be mapped.

4. The spatial conflicts must be identified and resolved. This can be achieved by various means: generalisation, change of symbols and/or their size and/or relocation.

5. The final stage of the expert system would be to evaluate the map by measuring it´s effectiveness.



FUNCTION REQUIRED	REASON FOR INVESTIGATION	POTENTIAL PROBLEM	POSSIBLE SOLUTIONS
1)Is a point on a line?	Points must not be obscured by lines.		MASK SIZE SYMBOL TYPE MOVEMENT
2)Is a point in polygon?	Points must not be obscured by polygon symbol.		CHANGE SYMBOL TYPE MASK
3)For any two lines, how much segment overlap is there?	Contiguous lines can obscure information.		MOVEMENT SIZE SYMBOL
4)For any points, are they clustered?	Points must not obscure one another.		GENERALISE SIZE EXPAND
5)Number of and distance between points in cluster.	If large number then spatial separation method must not be used.		
6)Identify optimum centroid of cluster.	To spread points without destroying local spatial integrity.		
7)With area of polygon and number of occurrences, calculate free map area.	FMA used to evaluate complexity of maps, and use of symbols.		

FEATURE CODE	1	2	3	4	
FMA	17	25	19	39	100
NO. OF OCCURENCES	5	6	4	4	19
MEAN Fc SIZE (FMA/occ.)	3.4	4.2	4.8	9.8	

Figure 1 Evaluative Functions, Problems and Solutions

The central difficulty of automating the map design process is in quantifying these tasks, which are presently performed by the cartographer. This is essentially a cartographic problem, not a computer one. This contribution is a start in that direction and is based around a discussion of algorithms for identifying and resolving spatial conflicts.

ALGORITHM DEVELOPMENT OF EVALUATIVE TECHNIQUES

Optimising the design of a map includes aspects such as balance, clarity, and contrast. These aspects are governed by factors such as scale, content, and the size of the finished product. Techniques are required to evaluate the data spatially in order to determine the nature of the conflict, and having considered the possible solutions, to resolve that conflict. Figure 1 shows some types of evaluating techniques required. It demonstrates the type of conflict they will detect and shows possible solutions to the conflicts, from which an expert system might choose the optimum, given the constraints relating to those offending elements.

It is worth stressing that any conflict in a map is essentially a spatial problem. A monochrome example will be used to demonstrate this: let us suppose a map contains areas symbolised using the same tone area fill as a point symbol, such that any symbol falling within that area will be indistinguishable. A conflict will only occur if there are point symbols that must be represented within those areas. Otherwise there is no conflict.

Conflicts can also occur when there is a change of scale. Suppose a large pictorial symbol (a flag on a pole) is used to show the holes on a golf course. If the golf complex is large (or the number of holes few) then the use of pictorial symbols is satisfactory. If however there are a group of holes clustered together, or the map is produced at a smaller size then spatial conflicts will occur. Two facts should be apparent from the above illustrations; the first is that the database containing the information must be based on spatial proximity; no one item can be changed without due consideration of its impact on the rest of the map. Secondly, in order for an optimum solution to a conflict to be found, any one point (line or polygon) must ´know´ about its local environment and what other data lie in its immediate vicinity (it´s property list).

Spatial Proximity Data Base (SPDB)

Most cartographic data are digitised and stored in vector format. A great deal of research has been done on storing such information in such a form as to enable fast retrieval (specified according to area or feature - fc) from an efficient and compact storage space. The GIS requirements for a design, where the system (not the user) must identify the conflict, are however quite different.

The format and accuracy of both the data, and the database determine (to a large degree) the efficiency and ability of a system to determine and resolve spatial conflict. The structure of the database must enable the system to efficiently determine both the property list of any one feature, and the proposed symbol that will be used to represent that feature. A database based on spatial proximity was investigated by Matsuyama and coworkers (1984). The system must first search the database for spatial conflicts, and identify the components of each conflict. Identification of those components would not just include measuring the Euclidean distance between each, but also parameters such as the degree of enclosure (the amount by which a group of features are enclosed by a line).

A cluster analysis program has been written which uses least squares as a measure of distance between cases and average linkage to estimate the position of groups of cases in relation to other groups or individual cases (see Mather, 1976). The least squares method gives a Euclidean distance by finding the square root of the sum of the squares of the difference between the variable scores (x, y) of each pair of symbols. Average linkage has been selected as a reasonable method of clustering, because it attributes importance to the group rather than to extreme individuals. The program determines which items are clustered, how they are clustered (number of cases at each cluster level) and records clusters containing ´offending´ points, where ´offending´ is defined as points which are within a minimum distance of their neighbour. The array of points in Figure 2 are analysed and the results are pictorially shown in Figures 3. Figure 4 shows the ´spatial dendrogram´ generated by the cluster analysis program. Such a ´tree´ can be envisaged as diagramatically representing the database; as one moves down the tree, the system can automatically identify clusters, their components and proximity.

```
Points  2 and  3 are   0.22 apart with  2 points in the cluster at level 1.
Points 10 and 11 are   0.28 apart with  2 points in the cluster at level 2.
Points  2 and  4 are   0.36 apart with  3 points in the cluster at level 3.
Points  6 and  7 are   0.45 apart with  2 points in the cluster at level 4.

   4 offenders recorded.



Number of symbols:     20
Number of variables:     2

        Cluster Analysis Tree Diagram
        ============================
Dendrogram of the distance similarity matrix -NNA.DAT

  1   --------!
              !
  2   -!      !
       !-!    +---!
  3   -! +----!   !
         !        +--!
  4   ---!        !  !
                  !  !
 10   --!         !  +-!
        +---------!  ! !
 11   --!            ! +-!
                     ! ! !
  5   -------------! ! ! !
                   ! ! ! !
  6   ----!        +-! ! !
          +--------!   ! !
  7   ----!            ! !
                       ! !
  8   ---------!       ! !
               +-------! !
  9   ---------!         !
                         !
 12   -------!           !
             +--------!  !
 13   -------!        !  !
                      !  !
 14   -----------!    !  !
                 !    !  !
 15   -----!     !--! +-!!
           +-----!  ! ! !!
 16   -----!        !-! +!
                    !   !
 17   ----------!   !   !
                +---!   !
 18   ----------!       !
                        !
 19   ------!           !
            +-----------!
 20   ------!
```

Figure 4 Output from the cluster analysis algorithm

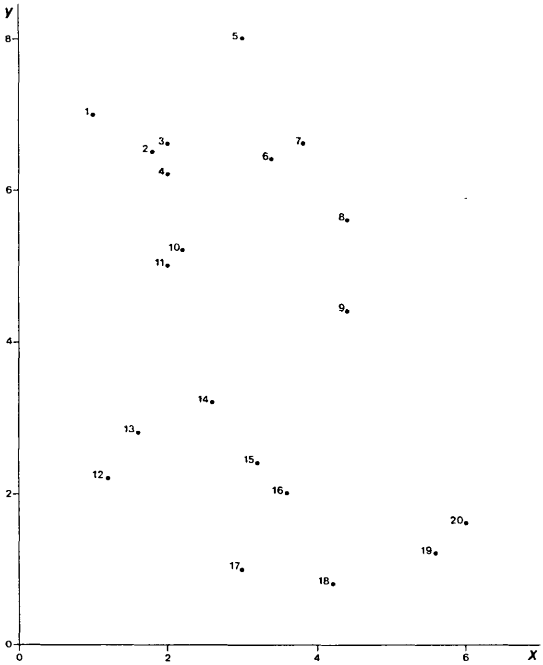


Figure 2 Randomly generated points

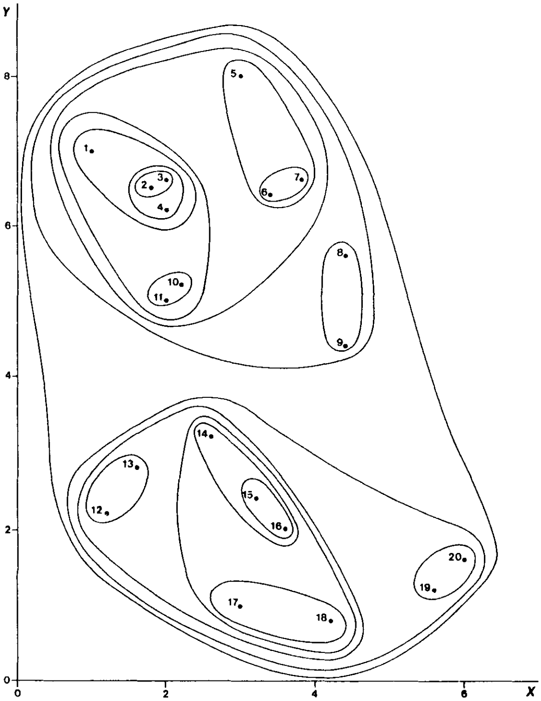


Figure 3 Clustering levels derived from cluster analysis algorithm

Proportional Radial Enlargement

As previously identified, there are a number of possible methods for resolving the problem of clustered data (Figure 1). One of those methods is to locally separate the points. It is not possible to separate the data by considering pairs of points, since the solution by movement of one pair, may infringe on adjoining data. Thus cluster analysis is used to determine how many points make up the clustered data. Any algorithm used to resolve clusters must satisfy the following objectives:

1. Their local spatial relationship much be preserved. Within the group of clustered points, the relative position of a point in relation to any other point must remain the same.

2. Whilst clarifying such information, the points must be moved a minimum distance in order to conserve spatial integrity. The amount of movement is determined by the initial proximity of points, and the size of the intended symbol.

One method that maintains ´shape´ is proportional radial enlargement; this involves selecting a centre and moving all the points away from the centre a distance, d, such that d is proportional to the original distance from the centre to that point. Where no other infringing data exist, the centre can be taken as the centroid of all the points (having equal ´mass´ or importance). The centroid, by definition will automatically gravitate towards the most dense part of the cluster, thus moving the majority of points the least amount. The shape of the group is preserved regardless of the position of the centre of radial enlargement (law of similar triangles). Figure 5 shows three such enlargements, each with different centres of enlargement. In Figure 5a the centre is the centroid of the points. Figure 5b shows another group of expanded points. Again the centre is taken as the centroid of the group. Some of the points now lie on the other side of the line. This is cartographically unacceptable; one solution is to alter the position of the centre. This could be done by altering the ´masses´ of selected points, which would effectively change the position of the centroid.

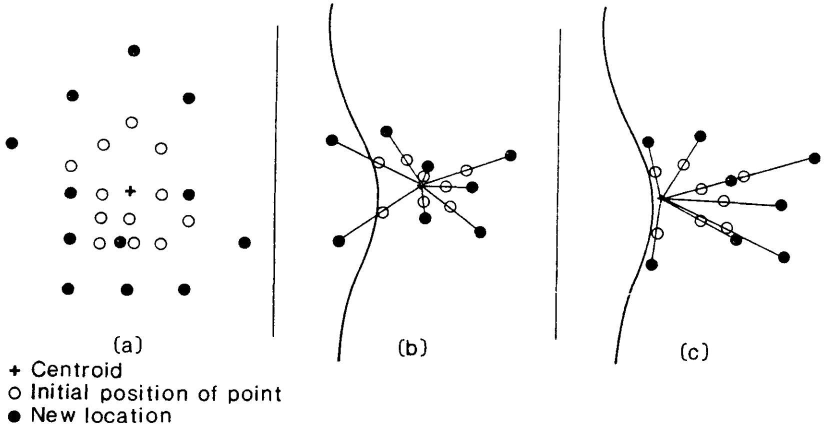


Figure 5 Radial Enlargement of Groups of points.

In all these cases the positions relative to each other are not compromised. There are however worst case situations where this solution is not appropriate (see figure 6), and alternative solutions (such as generalisation) must be considered.

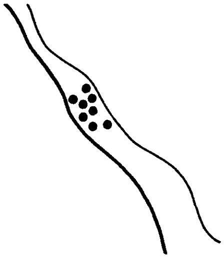


Figure 6 Worst Case

In a situation where spatial integrity was not important, (for example in a map showing train or bus routes) a high amount of total movement would be acceptable. In a map where spatial integrity was crucial then alternative methods must be used to clarify the data, such as a change of scale.

However when proportional radial enlargement is used, there is a loss of spatial integrity between the localised clustered features and the rest of the map (global features). One method used that is a compromise between the conservation of local spatial integrity and global/local blending is to use Gaussian distributions (optimally fitted to each expansion of points) to determine the ratio of movement. Thus the ratio of movement gradually decays towards the fringes of the cluster. Figure 7 compares two expansions. Figure 7a is a radial enlargement by a fixed factor of enlargement k, such that the distance of movement is proportional to the original distance from the centroid to the point. In Figure 7b the Gaussian decay curve is used to determine the value k, which decays to 0 as the distance from the centroid tends to infinity.

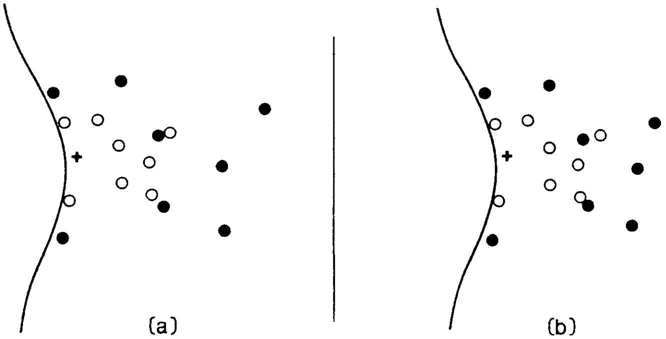


Figure 7 Proportional and Gaussian Radial Enlargements

Map Evaluation

At all stages there would be a need to evaluate the success of the design. Various parameters must be measured such as the total amount of movement of objects, changes in base data and re-symbolisation. The thresholds of these parameters will depend on map type, audience (those who will use the map) and size of finished product, and on the spatial properties of the data (for example symbols used to show an even distribution may not be appropriate for showing a clustered distribution). If the thresholds are exceeded, then an expert system might be used to decide on one or some of the factors that can be altered to reduce the threshold. These various alternatives include the reduction of symbol size and/or base data, and generalisation.

CONCLUSION

A human cartographer must first be able to identify conflicts in map design, and have at his disposal methods of resolving those conflicts. An essential prerequisite for a cartographic expert system must be equivalent methods for identifying and resolving spatial conflicts.

It is apparent that the solution to any spatial conflict involves first identifying the amount and types of data that lie in the immediate vicinity. Methods for efficiently searching the database for conflicts will depend on the format of the database. Only once the components of the conflicts have been identified can an optimum choice be made from the ´possible solutions´ (see Figure 1).

The approach outlined in this paper differs from other attempts to automate map design in that it views the problem of map complexity as a whole, not as a set of sequential design stages. No cartographer makes a map by selecting the data, symbolising, placing, adding text and finally drawing the key; rather they modify their decisions during the design, both at the global level (in deciding the maximum acceptable information content) and at a local level (where information is obscured because it is clustered together). If expert systems are to draw maps (and not just technical drawings) then they must have the equivalent cartographic senses; eyes with which to discern, knowledge with which to make decisions, and an inference system that enables it to change decisions during the design phase.

If a quantitative model can be developed that can optimally resolve clusters of points, it should be feasible to extend the method to include the similar types of problems found in line labelling and text placement situations. Further research is required specifically to algorithmically determine the degree of contiguity between lines (the amount of overlap), and calculate a value for the degree of enclosure. Alternative methods for resolving worst case clusters must also be identified (see figure 6).

REFERENCES

Barr, A. and Feigenbaum, E.A. 1982. The Handbook of Artificial Intelligence, 3 Volumes. London, Pitman.

Green, N.P., Finch, S., and Wiggins, J. 1985. ´State of the Art in Geographical Information Systems´ Area 17(4) pp295-301.

Mackaness,W.A., Fisher,P.F., and Wilkinson,G.G. 1985. The Design of a Cartographic Expert System Final report to NERC Contract F3/G6/304

Matsuyama, T., Hao, L.V. and Nagao, M. 1984. ´A file organisation for Geographic Information Systems Based on Spatial Proximity´ Computer Vision Graphic and Image Processing 26 (3). pp303-318

Mather, P.M. 1976. Computational Methods of Multivariate Analysis in Physical Geography London: John Wiley.

Monmonier, M.S. 1984. Geographic information and cartography. Progress in Human Geography 8, 381-391.

Morrison, J.L. 1980. Systematizing the role of "feedback" from the map percipient to the cartographer in cartographic communication models Paper read to the International Cartographic Association, Tokyo.

Page, E.S. and Wilson, L.B. 1978. Information Representation and Manipulation in a Computer 2nd edition. Cambridge University Press.

Peuquet D.J. 1984. ´Data Structures for a Knowledge Based Geographic Information System´ Proceedings Of The International Symposium On Spatial Data Handling Aug 20-24 1984 Zurich, Switzerland Vol 2, pp372-391.

Smith, T.R., and Pazner, M. 1984. ´Knowledge Based Control of Search and Learning in a large scale GIS´ Proceedings Of The International Symposium On Spatial Data Handling Aug 20-24 1984 Zurich, Switzerland Vol 2, pp498-519.

CALCULATING BISECTOR SKELETONS USING A THIESSEN DATA STRUCTURE

Robert G. Cromley
University of Connecticut
Storrs, Connecticut 06268

ABSTRACT

An important construct for analyzing the shape and structure of a polygon in Euclidean space is its bisector skeleton. A bisector skeleton partitions the area of a polygon into subpolygons that are closer to one edge of the polygon or its internal linear extension than to any other one. While bisector skeletons are unique in their form and application, their cartographic elements are topologically equivalent to those of a Thiessen diagram. Consequently, procedures for calculating Thiessen diagrams may be adopted for calculating bisector skeletons, just as Thiessen procedures can be applied to the problem of Delaunay triangulation. This paper presents an algorithm for constructing bisector skeletons using a triangle data structure and the form of a procedure for identifying Thiessen diagrams within a convex boundary.

INTRCDUCTION

An important class of cartographic problems is related to the partitioning of space based on proximity criteria. One proximity problem given a known point distribution, is to delineate the set of points on a surface that is closer to one known center than to any other points. This problem of constructing a Thiessen diagram has many applications in economic geography, quantitative techniques, and cartography. Another proximity problem is that of constructing the bisector skeleton of a given polygon. Bisector skeletons partition the internal area of a polygon into subpolygons that are closer to one edge of the polygon or its internal linear extension than to any other edge and its extension (Brassel, Heller, and Jones, 1984). The inclusion of internal edge extensions in its definition distinguishes bisector skeletons from earlier continuous skeletons (Montanari, 1969) and gives them their strictly linear appearance (see Fig. 1). It should be noted that while Thiessen polygons are convex, bisector polygons may be either convex or concave (again see Fig. 1).

Algorithms for analytically delineating Thiessen polygons have received more attention (Rhynsburger, 1973; Shamos, 1977; Brassel and Reif, 1979) than the newer developed problem of constructing bisector skeletons. Recently, an algorithm for identifying and storing a Thiessen diagram *within a convex boundary* has been proposed based on a triangulation data structure (Crcmley and Grogan, 1985). The purpose here is to apply the design of this procedure to the problem of calculating bisector skeletons after showing the topological equivalence of respective cartographic elements of each diagram.

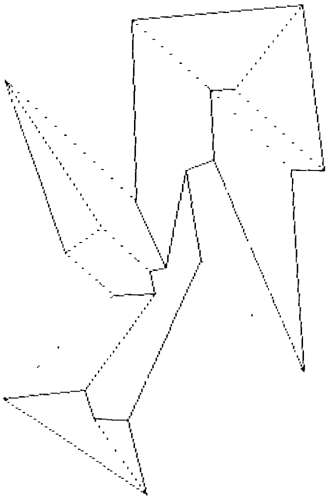


Figure 1. A Bisector Skeleton

CONCEPTUAL BACKGROUND

There is a clear analogy between the components of the bisector skeleton problem and identifying a Thiessen diagram within a convex boundary. For the Thiessen problem, a set of n points is given in a plane bounded by a convex polygon defined by a set of m edges. Each edge of the bounding polygon is a line segment connecting two boundary vertices; thus, the polygon is alternatively referenced by m vertices (Cromley and Grogan, 1985). It is assumed that the bounding vertices are sequentially numbered in a clockwise direction so that the Thiessen diagram is always to the right as one moves around polygon boundary. For the bisector problem a polygon composed of n edges is given in a plane. Each edge of the polygon is a line segment connecting two boundary vertices; again the polygon is alternatively referenced by these n vertices. It is also assumed that the polygon's vertices are sequentially numbered such that as one moves from vertex to vertex, the area of the polygon lies to the right of the connecting edge.

The set of n points are used in the Thiessen problem to generate convex polygons that are nearer to one point or

Thiessen centroid than to any other centroid. Likewise, the set of n edges of a polygon are used in the bisector skeleton problem to delineate subpolygons within the given polygon that are closer to one edge or its interior extension than to any other edge or its corresponding extension. Given an identical function within the context of the respective problem, each Thiessen centroid, C_i, is equivalent to each polygon edge, P_i.

The problem of identifying a corresponding polygon for each centroid is equivalent to finding a set of p points that are equidistant and closest to three centroids (Fig. 2); these points are called Thiessen vertices. Similarly, a skeleton vertex is a point equidistant and closest to three polygon edges or their interior linear extension (Fig. 3) (Brassel et al, 1984). Thiessen vertices and skeleton vertices are also equivalent as they represent junctions along the perimeter of local polygons where the generating centroids (polygon edges) change neighboring centroids (edges). Additionally, each Thiessen vertex that lies on the convex bounding polygon is called a boundary Thiessen vertex while the others are known as interior Thiessen vertices. Similarly, skeleton vertices will either lie in the interior of the polygon or on its boundary; in the latter case, the set of boundary skeleton vertices is identical to the original set of n polygon vertices.

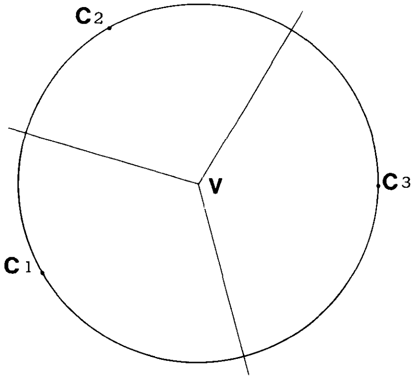


Figure 2. A Thiessen Vertex and Its Nearest Centroids

Finally, a Thiessen edge, E_i, is defined as the locus of points equidistant and closest to two centroids. A Thiessen edge will connect two Thiessen vertices that share two nearest centroids. Thiessen edges connecting two boundary Thiessen vertices are known as boundary Thiessen edges. For a bisector skeleton, a skeleton edge, S_i, is the locus of all points equidistant and closest to two polygon edges.

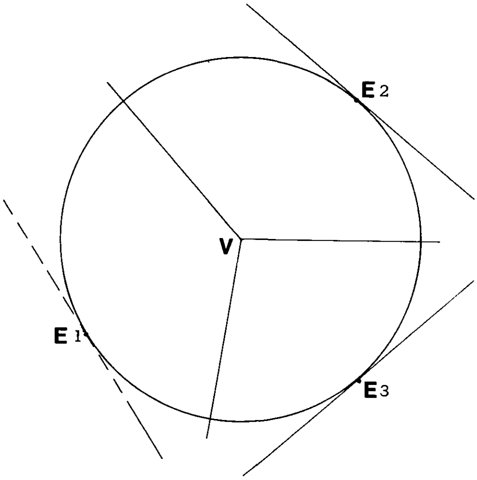


Figure 3. A Skeleton Vertex and Its Nearest Edges

Skeleton edges will connect skeleton vertices that share two closest polygon edges. Therefore, skeleton edges that connect two boundary skeleton vertices are also called boundary skeleton edges. In this case, the set of these boundary skeleton edges is the same as the set of polygon edges; in other terms, a polygon edge in the bisector skeleton problem is equivalent to two different elements in a Thiessen diagram with a convex boundary: a Thiessen centroid and a boundary Thiessen edge.

Additionally, one centroid is called a Thiessen neighbor of another centroid if the two centroids' polygons cobound the same Thiessen edge. It should be noted that some edges may connect two unique vertices that have the same cartographic location which gives the visual impression that one vertex has more than three nearest centroids and that some centroids share only a common vertex rather than an edge. However, only the centroids sharing the zero length edge are neighbors of each other (Cromley and Grogan, 1985). All centroids with a boundary Thiessen edge are neighbors of an imaginary background centroid, C_{n+1}. Analogously, one polygon edge is a skeleton neighbor of another if the respective subpolygons share a common skeleton edge. By definition, each polygon edge will be a skeleton neighbor of an imaginary background edge, P_{n+1}.

It is important to enumerate each Thiessen (skeleton) vertex and edge as an exact number of them exist as a function of the number of centroids (polygon edges). Cromley and Grogan have shown that a Thiessen diagram with n centroids will have 2(n-1) vertices and 3(n-1) edges. Similarly, a polygon with n edges will have 2(n-1) skeleton vertices and 3(n-1) skeleton edges. While it is unknown how many boundary Thiessen vertices there will be, it is always the case that there are exactlv n boundarv skeleton vertices and therefore n-2 interior skeleton vertices.

It is also imnortant to enumerate each Thiessen or skeleton vertex because respective vertex reference files can be constructed based on the tonological relationshins between vertices and centroids (polygon edges). Because a Thiessen diagram is the dual of a Delaunay triangulation, a Thiessen data file is based on Elfick's triangle structure (Cromley and Grogan, 1985). Each record of the file contains six entries corresponding to the neighborhood information of each unique vertex and two entries for its coordinates. The first three neighborhood values contain the references of the three adjoining vertices recorded in a counterclockwise order. The next three entries are the reference values of the corresponding centroid (polygon edge) whose generated polygon is the right-hand neighbor of the edge connecting the current vertex to an adjoining vertex. Thiessen or skeleton edges are not retained in this file as they are line segments connecting vertices and would be redundant information. Once all 2(n-1) records have been completed, a digital representation of a bisector skeleton or a Thiessen diagram is complete.

ALGORITHM DESIGN

Cromlev and Grogan have nresented a two stage method for constructing the vertex reference file for a Thiessen diagram. In the first stage, all Thiessen boundary vertices are enumerated by walking around the outline of the bounding polygon in a clockwise direction. As each boundary vertex is found, its three centroid neighbors and the two adjoining vertex neighbors that are also boundary vertices are identified. Only the third vertex which is an interior vertex remains to be identified. In the second stage, the interior vertices are found by moving along the boundary of each individual Thiessen polygon in a clockwise manner. This process starts by first enumerating those Thiessen polygons that have a boundary Thiessen edge and then continues in an inward spiral until all polygons and vertices have been found. As the boundary of an individual polygon is completed, its generating centroid is removed from the list of potential centroid neighbors for new vertices.

A similar procedure can be anplied to the bisector skeleton problem. In this case, the first stage is trivial as the set of boundary skeleton vertices is the same as the given set of polygon vertices. The second stage is also less complicated as the spiral process terminates when the last subpolygon having a boundarv skeleton edge is completed as there are no subpolygons in a bisector skeleton that are completely interior to the bounding polygon.

While the overall design of the bisector skeleton procedure is the same as a Thiessen procedure, there are many technical details that differ. First, the edge that partitions the subnolygons is formed by the bisector of the angle between two polygon edges or their extensions rather than the nerpendicular bisector between two centroids. Second, Brassel and Reif's circle test for finding centroids that are closer neighbors cannot be used. Instead, the third neighbor (a polygon edge in this case) for each interior vertex is found by sequentially testing each polygon edge in a clockwise order. A half-plane test is used to determine if the last vertex lies in the same half-plane as the edge being tested or in the half-plane of the current potential edge. The current potential edge is updated whenever the vertex is in the half-plane of the new edge. Finally, as one proceeds around the boundary of each subpolygon, only those edges that are subsequent in the clockwise order of the last polygon edge neighbor need to be tested as potential neighbors for new interior vertices of the current subpolygon. This algorithm has been implemented in FORTRAN 77 and used to construct Fig. 1.

SUMMARY

The algorithm presented here has shown that calculating bisector skeletons is analogous to that of calculating Thiessen diagrams. Although many cartographic entities have very different forms and functions, their digital form is often quite similar. This enables digital methods to be more integrated than their manual counterparts.

REFERENCES

Brassel, K., M. Heller, and P. Jones, 1984, The Construction of Bisector Skeletons for Polygonal Networks: Proceedings, First International Symposium on Spatial Data Handling, Vol. 1, pp. 117-126.

Brassel, K. and D. Reif, 1979, A Procedure to Generate Thiessen Polygons: Geographical Analysis, Vol. 11, pp. 289-303.

Cromley, R. and D. Grogan, 1985, A Procedure for Identifying and Storing a Thiessen Diagram within a Convex Boundary: Geographical Analysis, Vol. 17, pp. 167-175.

Elfick, M., 1979, Contouring by Use of a Triangular Mesh: The Cartographic Journal, Vol. 16, pp. 24-29.

Montari, U., 1969, Continuous Skeletons from Digitized Images: Journal of the Association for Computing Machinery, Vol. 16, pp. 534-549.

Rhynsburger, D., 1973, Analytic Delineation of Thiessen Polygons: Geographical Analysis, Vol. 5, pp. 133-144.

Shamos, M., 1977, Computational Geometry, Ph.D. dissertation, Yale University, 1977.

AREA MATCHING IN RASTER MODE UPDATING

J.P. GRELOT, P. CHAMBON and F. MARCHE
Institut Géographique National
136 bis rue de Grenelle
75700 PARIS - FRANCE

ABSTRACT

Cartographic data bases updating is a real challenge for the maintenance of the homogeneity and reliability. The choice has to be made between interactive editing and automatic procedures. An attempt has been made by IGN France on the occasion of a land-use inventory updating. More precisely, the aim was to adjust a new delineation of land use areas to an old one. Digitization is performed in raster mode through a scanner. After skeletonization, control points are interactively identified in order to calculate a global transformation which is applied to the vectorized boundaries, and eliminates main distorsions due to variations in drawing materials. Then the boundaries are processed in raster mode in order to correct new lines which are exactly adjacent to old ones, or which define with an old line a micro-area. The results seem rather good but introduce local distorsions due to the very local approach of the process. To avoid such distorsions, we have to improve the software in using shape considerations in replacing very local operations by more regionalized operations. It is an interesting example of the successive or simultaneous use of several data structures to solve an updating problem.

INTRODUCTION

When a cartographic compilation is made to follow the evolution of a phenomenon, the updating process must be forecasted. Digital cartography allows us to make it easier, solving the problem of bad conservation of documents. However, in the case of a land use inventory, updating informations have to be collected carefully, because they cannot be combined directly with the device that will be used to produce the map : the file on the computer. Therefore, problems appear when trying to superimpose the updating files with the old ones, producing poor cartographic results, incompatible with the rules for representing the phenomenon and which do not have any real meaning. This problem of misregistration of digital areal information has been solved at IGN France for a project concerning a series of maps.

UPDATING A LAND USE INVENTORY

The area matching project began three years ago when we had to update a land use inventory known as "the French Littoral Inventory", which has been designed as a management tool with a periodical depiction of the area of interest (Grelot 1982). Let us have a few words about it. The aim is creating some objective data on the coastal area in order to help in leading a protection policy in this area. You may regard France as a Far-East country, but you also may know that its coastal area gathers about one tenth of the population and has a major touristic appeal for many European people coming there from cloudy and cold countries. The basic tool we designed is a land use digital map at 1:25 000 scale made from aerial photographs interpretation and from subsidiary data compiled by local administrations (Grelot and Chambon 1986).

We defined an initial coverage with about 400 000 polygons and the legend consisted in about thirty land-use categories which had been selected after a long discussion involving many people from local and governmental administrations as well as from technical organisations. However, with the availability of the initial map coverage, some people requested to re-define the legend. In some cases we could subdivide some categories and it was only a change in the attributes allocated to the polygons. But in many cases the result was a new set of polygons we had to substitute to the former one. And everyone knows that modifying boundaries is much more difficult than allocating attributes.

On this background makes the concern in terms of area matching easily understandable. We had to update the data files in two ways : the first one was due to real changes in the countryside during the five-year period and the second one was due to a new classification mainly in urban areas and also in agricultural areas. Because we had a huge number of polygons, we decided to use automated processes instead of interactive ones and we developped an area-matching facility.

However a major hypothesis was absolutely wrong. We thought that the main reason for distorsion during the drawing was graphic uncertaincy which we agreed upon because of the average size of polygons. In fact we had very bad misregistrations due to three main sources : (a) dimensional variations in the base map used as the background for drawing the boundaries, (b) poor drawing quality, and (c) graphic uncertaincy. In that respect a scale factor and some local processing were not sufficient for matching archived and up-to-date boundaries, and we had to design a new software package.

The main requirements for the matching package were :
(a) automated processing with only minor interactive tasks and no editing;
(b) input and output data files in raster mode ;
(c) defining then computing local geometric distorsions ;
(d) preserving old boundaries ;
(e) local capture of up-dated boundaries;
(f) automatic re-allocation of land-use attributes.

THE AREA MATCHING PROCESS

This process deals only with zonal information. These data constitute files in raster mode, made of homogeneous and connex zones where one or more attributes are given.

The initial data set is used as a reference and considered as geometrically exact. The updating document contains only the new information. The boundaries of the new closed areas are then hand-drawed and digitized on a drum scanner. The features are skeletonized and some editing is made on a graphic workstation in order to avoid errors due to digitization and to close every area. This file is then superimposed to the old state one, in raster mode, and the operator can see the quality of registration. Homologous points (20 to 30) are identified. The coordinates of this points are taken interactively in order to calculate a global deformation between the old and the new boundaries.

Here begins a series of functions which are processed automatically one after the other.

The global deformation is represented by a biquadratic polynom whose coefficients are computed by the least squares adjustement method. By applying this polynom to the new boundaries after vectorization, we make them fit the old ones, with the best accurary on the control points.

A check plot is made after this transformation showing :
a) the boundaries of the old areas.
b) the boundaries of the new areas (having changed between the two states).
c) the same boundaries as b), having undergone the deformation.

We can also see the position of the control points on this plot, made on a large format electrostatic colour plotter. That is where we can notice that the final boundaries are well fitted to the old ones near the control points. On the other hand, where fewer control points have been identified, a misregistration remains. But everywhere, this error is reduced to less than half a millimeter, though it could exceed two millimeters initially. If this is not the case, we have to take other control points an remake the process.

At this stage, we can see when superimposing the new and the old boundaries that only local errors are still present and are represented either by small areas (areas having a small surface) or by prolate areas (one or some pixels wide). These parasite zones do not have any real thematic significance and must be suppressed, the aim always being to keep the old boundaries when there is a conflict. These suppressions are made by a special software which has several effects :
a) It detects areas whose surface is less than one square millimeter, or which are less than 4 pixels (0,5 millimeter) wide,

b) an iterative algorithm forces the new boundaries to be just near the old ones (distance is exactly one pixel) along these micro-areas,
c) when such sticked boundaries are encountered, they are merged in favour of the old ones.

Another check plot is made at this stage which shows the final results ; on this plot where the areas are labelled, we can also identify corrections to be made, due to other reasons.

The whole process requires approximately 5 computing hours, and some additional time to check the plots. During this time, up-to-date attributes are interactively given to the new areas defined by the digitized boundaries (only those having changed since the old state). After the matching procedure, the old and the new boundaries are superimposed, defining new areas which all have a thematic significance. These new areas receive automatically the old and the new thematic attributes. That is where we stop the matching process, maps being made as for an usual work (Grelot and Chambon 1986).

ADVANTAGES AND DRAWBACKS OF THE METHOD

The matching procedure merges neighbour boundaries, i.e. made with adjacent pixels, and eliminates small linear areas which look like spaghettis with two parallel boundaries, one old and the other updated. The surface of such areas to be eliminated is less than a given threshold value.

Of course this automatic procedure is not perfect. The tolerance after the distorsion is about two to three pixels which means about a quarter millimetre. There may be some misinterpretations with linear features such as rivers and major roads and railroads. They are removed during the final editing stage.

You may have noticed that any ckecking is made from a graphic plot. In other words it means that only those defaults with graphic significance can be seen. In fact it is not realistic to display every file through small windows with a large zooming factor in order to detect any default. And we all know some defaults which do exist in raster boundaries after skeletonizing hand-drawn lines : there are some random waves and on the other hand some systematic smoothing upon angular shapes. The matching process is not responsible for them and does not try any improvement.

The matching process is perfomed on small areas with a pixel-by-pixel approach. Not only during the comparison stage, but also during the decision and drawing stages. And there is a need for a smarter solution in the area separating a to-be-changed from a to-be-kept boundary set. It means that the local approach induces defaults in lines because the lines are considered as sets of adjacent pixels and not as features with particular shapes. Those defaults are made of very short segments joining the old and the new locations of the two parts of a boundary : they do not create graphic troubles but they are logical errors in the depiction of lines.

Finally the procedure :
(a) adjusts the updating file to the old one through a smooth distorsion to reduce misregistrations to the order of magnitude of graphic uncertaincy ;
(b) matches the boundaries pixel by pixel ;
(c) increases some defaults of raster files such as random waves and smoothed angles.

I do not think that a vector-mode process would give better results. On the contrary I do consider that this process successfully combines vector-mode stages and raster-mode stages. It proves that the link between data structures and processes is more important than the link between structures and models, and how interesting is the capability for using several data structures in the very same data base depending on applications.

Let us have a come-back to look for improvements. First the graphic quality could be improved. A land-use pattern has not the same characterics as a geological one : it is more related to the land ownership and to parcels with straight boundaries and precise corners. There is a close connection between the geometric characteristics of a boundary and the nature of adjacent areas, in other words between geometric and semantic describers. As an application for our purpose we should look for algorithms enhancing the raw data set and getting more straight boundaries and sharper angles. I can add a comment : as this geometric processing is due to the very nature of areas, it can only be performed after attribute allocation.

Second point : the boundary patterns are not similar in the two files. It is evident because the second data set is only drawn for updating the first one, but it has to be reminded and it has some consequences. The main one is that we cannot select a set of points (for instance the nodes) within the updating file and look for the geometric distorsion which maximizes the correlation with a corresponding set of points from the old file. It also means that the result of the distorsion is very dependent upon the interactive selection of control points and of some bad effects of skeletonization. It would have been useful to get a network of fiducial marks displayed on both lineworks for automating and improving the mathematical distorsion.

Third point : a major quality achievement should be made when explicitly considering the shapes of features. This is not easy. It means that the local processing which now removes points should be replaced by only a comparison which determines a displacement to be applied on a boundary or an entire set of boundaries. There are some requirements for making this : for instance, the nodes and corners of polygons and some geometric characteristics of the shapes have to be known in order to avoid large modifications of shapes. The displacements have to be designed as a mechanical stress upon a solid network.

CONCLUSION

I think that these three points are the main directions for improving the area matching procedure. We have seen how fruitful is the capability for using several data structures during a particular process. We also have seen that local processing has to be complemented with a more global approach using the geometry of shapes.

These remarks may look pessimistic. Our process does not work as perfectly as we could expect, but it actually works. We use it for superimposing data sets on the same area related to the same kind of phenomenon such as a land-use updating or related to different matters such as land-use and soils.

A lot of interactive editing is replaced by an automatic process which efficiently eliminates noisy areas created when superimposing the data sets. And this was the purpose: any cartographic superimposition or updating induces a lot of matching work, and the digital challenge is not performing the matching on editing workstations but automating it as much as possible.

REFERENCES

Grelot, J.P., 1982, An Inventory of French Littoral: Auto-Carto 5 Proceedings, pp. 367-373.

Grelot, J.P. and Chambon, P., 1986, Up-dating a Land Use Inventory : Auto-Carto 7 Proceedings, to be published.

POLYGONIZATION AND TOPOLOGICAL EDITING AT THE BUREAU OF THE CENSUS

DAVID MEIXLER
ALAN J. SAALFELD
BUREAU OF THE CENSUS
WASHINGTON, D.C. 20233

ABSTRACT

In 1983 and 1984, the Bureau of the Census developed a computer program to polygonize digital map data (organize linear feature information into polygons) and to validate the topological and geometric correctness of the nodes, chains and polygons. The program evolved from an earlier planar sweep program, and eliminated many of its geometric dependencies. The program runs in both an insertion and edit mode. In the insertion mode, the program is used to generate the elementary topological polygons for over 50,000 maps covering nearly 3,000,000 square miles of the contiguous 48 states. This insertion process is a necessary component in the joint United States Geological Survey-Bureau of the Census project to produce the National Digital Cartographic Data Base. At critical points in the work flow, the edit mode is used to verify the underlying topological soundness of the file structure.

INTRODUCTION

The TOPOLY program evolved from an earlier planar sweep program and still retains some of its characteristics. A planar sweep program labels each unique polygon in a graph by visiting each node in the order it would be visited by a line sweeping across the graph [Nievergelt and Preparata, 1982].

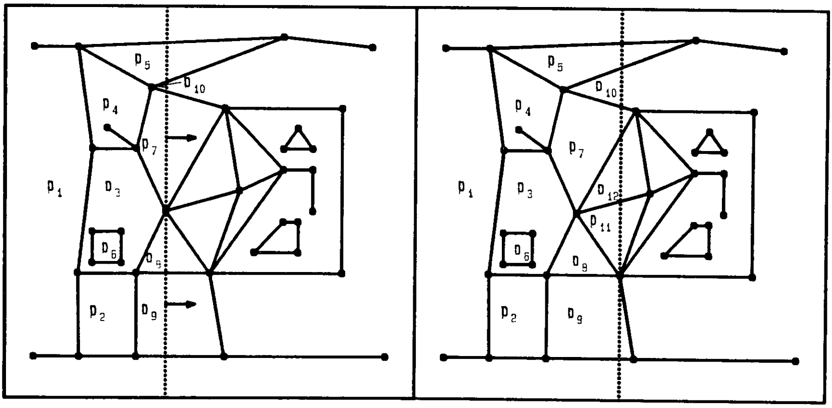


FIGURE 1 PLANAR SWEEP ALGORITHM ILLUSTRATED

When at a node, the algorithm will ensure that any new polygon labels placed upon the graph are consistent with existing labels previously applied. The planar sweep algorithm keeps track of the regions currently under this line and can also check for undiscovered intersections of the lines or resolve discontiguous parts of the graph (islands) that are uncovered as the line sweeps over the graph.

The TOPOLY program is more efficient with the Census Bureau file structure and needs no overhead for keeping track of the regions under the sweep line. Since intersection checks are done in a separate process at the Census Bureau, an intersection testing capability was not needed in the polygonization process. However, when the Census Bureau staff adopted the unordered choice of nodes on which to apply consistent polygon labels, they still needed processes to detect and resolve the disconnected complexes in the graph. The detection process fit with other polygon analyses being done. The resolution of the disconnected pieces became a new process. The program thus has four parts: the initial label placement on the sides of the chains; the analysis of the resulting polygons; the resolution of the discovered islands; and the modified Euler edit at the end of processing.

POLYGON LABELING

The insertion of polygon labels in the graph is a simple and fast process. All the nodes in the file are visited in whatever order they are stored in the file. The chains (1-cells) attached to them are collected and ordered by angle of emanation from the node; this ensures that the facing sides of adjacent chains are labeled consistently.

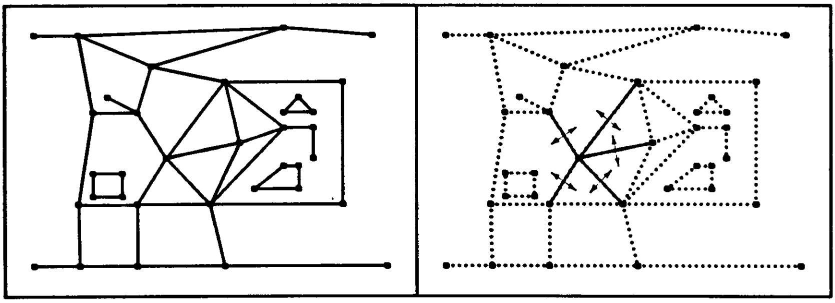


FIGURE 2 UMBRELLA EDIT
COLLECTING CHAINS ABOUT A NODE

If no label exists between the chains, a new label is created and inserted on both. If the labels are different, one of the labels is preserved and the other deleted. All chains having the deleted label are reassigned the preserved label. If the labels are already consistent, the next pair of chains is visited. These are the same rules applied in a planar sweep process. The important difference is that the nodes are visited independently of their position. Thus the program is called TOPOLY, for TOPOlogical POLYgonization.

The labeling process and the umbrella edit are equivalent. When running in edit mode, this section of the program reverts to an umbrella edit [White, 1984]. This edit examines the chains emanating from every node to ensure that the polygon labels on facing sides of the chains are consistent. For example, if angles of emanation of the chains increase when measured counter-clockwise, then the left polygon label on one chain should agree with the right polygon label on the next chain. Similarly, the left label on this next chain must agree with the right label on the subsequent chain. This edit treats each node as the hub of an umbrella and tries to ensure that the order of the emanating spokes (chains) is consistent with the labels of the area between them.

If this edit is done on every node of the graph, and each node is consistent, then the graph is guaranteed to be labeled consistently. However, this consistency of labeling does not guarantee a topologically sound file. The most obvious case of an unsound file would be the case where the entire file has the same polygon label. Assuming the graph has at least one cycle, then there must be more than one polygon. However, the umbrella edit will not detect two atomic polygons with the same label. It will detect the lack of consistency around individual nodes. Thus failure of the umbrella edits insures that the file is unsound, but passing it does not insure topological consistency.

POLYGON ANALYSIS

The major process in the polygon analysis phase is the Kirchoff routine. A Kirchoff analysis is done separately on each individual polygon in a file. The basic Kirchoff procedure counts the number of cycles and acycles in a set of chains [Prather, 1976]. In a complex graph, the number of independent cycles is a fixed number N. However, the N cycles themselves, i.e. the chains that constitute them, are not uniquely determined. In other words, there may be many sets of N independent cycles for a particular complex graph.

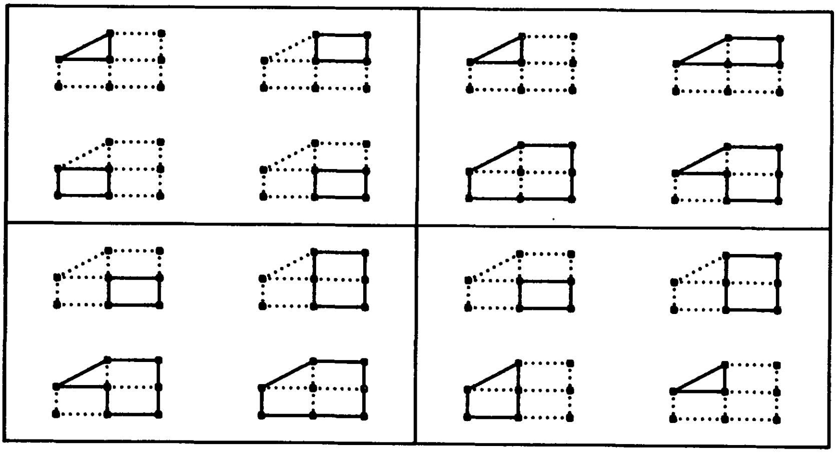


FIGURE 3 SETS OF INDEPENDENT CYCLES OF A COMPLEX GRAPH

For a topologically sound polygon, the cycles themselves are determined uniquely. In fact, every boundary chain belongs to a cycle. Nonboundary (internal) chains do not form cycles. Additionally, all boundary cycles are independent. One and only one cycle may be recognized as the outer boundary of that polygon. Any other cycles will constitute "inner boundaries", i.e. the chains that surround a "hole" in the polygon.

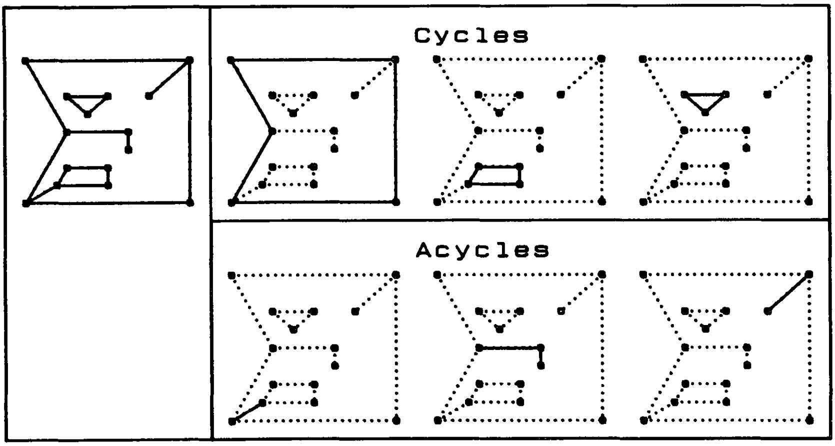


FIGURE 4 UNIQUE GENERATING CYCLES AND ACYCLES OF A TOPOLOGICALLY SOUND POLYGON

All of the chains with the same polygon label are submitted to a Kirchoff routine, and its analysis is the basis for confirming the integrity of the polygon. For example, internal (nonboundary) chains on a polygon are known from the fact that the left and right polygon labels for each of these internal (non-boundary) chains on a polygon are known from the fact that the left and right polygon for each of these chains are the same. Similarly, all boundary chains are known by the fact that the cobounding polygons have different labels. A Kirchoff routine analyzing this set of chains determines which chains form cycles and which are acyclic. All chains in the acycles are confirmed to be internal and all chains forming cycles are guaranteed to be boundary chains. The cycles are ordered, coordinates are extracted, and the information is sent to a routine that computes the area, perimeter, centroid, shape, and direction of traversal in one computation.

Orientation is used to identify inner and outer boundary cycles. The direction (either clockwise or counter-clockwise) of the cycle is compared with the side on which the polygon lies. When walking around the outer boundary of a polygon in a clockwise fashion, then the polygon must be on the right side. Similarly, a counter-clockwise traversal around an inner boundary keeps the polygon on the right side. If the direction changes, then the sides reverse. Thus to walk counter-clockwise around an outer boundary, keep the polygon on the left. The cycles returned by a Kirchoff analysis are examined to ensure consistent labeling on a specific side. The direction of the cycles also is examined to insure that there is one and only one outer boundary and that the area it encompasses is greater than that of all inner boundaries.

Another modification of the Kirchoff routine counts the number of discontiguous components in the chains associated with each polygon. This normally will be one. However, some polygons may have more then one component. The number of separate components in the graph is summed to be used later for the modified Euler edit described below. Also, the disconnected parts created in the initial labeling are recognized for the following island resolution process.

ISLAND RESOLUTION

During the labeling process, disconnected components will acquire a separate set of polygon labels. Except for the outer regions of the separate components, these labels are valid polygons with an outer boundary and zero or more inner boundaries. These valid labels can be ignored for island resolution. However, the outer region of each component must be resolved to agree with the other components of the graph. The polygon labels that form the outer boundary of a component of the graph will form what Kirchoff recognizes as an inner boundary and will have no outer boundary. The region encompassed by this boundary is sometimes referred to as a "hole" or island in the main component. The largest of these islands is recognized as the outer region of the entire map. Recognizing and labeling this area as the "first polygon" in the file is a requirement of the joint agreement with the U.S. Geological Survey.

Once this island is detected it is resolved to agree with the main component of the graph. If an intersection check has been performed already, then this island will fit entirely within a single polygon of the main component. The resolution itself is made by resolving a single point of the region to be within one polygon of the main component. This resolution is done starting with any polygon and chaining adjoining polygons to find the correct one. Wherever this point falls will have the entire outer boundary recoded to it. Island resolution is only necessary during the insertion mode. Its edit mode equivalent ensures that disconnected complexes are labeled to agree with the surrounding polygon. This is more convenient to do in the polygon analysis phase.

EULER EDIT

The final edit done on the file before the program terminates in both the insertion and the edit mode is a modified Euler edit. The Euler edit is based upon simple principles. If a single line chain is drawn and examined, it is obvious that there is a single chain, two endpoints and one surrounding region. In this example, the number of nodes and regions equals the number of chains, plus two. The Euler Theorem states that this numerical relationship is always true for a connected set of chains on a plane. This may be illustrated by iterative chain building. When a new chain is joined to an existing network of chains, one endpoint must begin at an existing node. If the other end of the added chain goes to a node that already exists, it will create a new bounded polygon inside an existing polygon or in the outer region. If the other end of an added chain does not link up with an existing node, it creates a new endpoint.

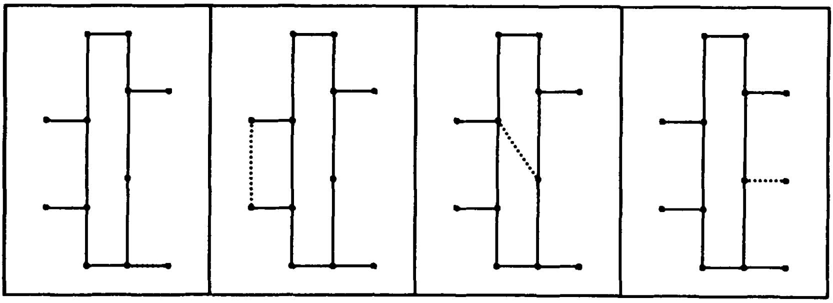


FIGURE 5 OPTIONS FOR ADDING A CHAIN
TO AN EXISITING CONNECTED NETWORK

Thus, after the basic relationships among the node counts, chain counts, and polygon counts are established, that relationship will remain constant as long as new chains are attached to the existing network of chains.

However, a cartographic database need not have all chains connected. This leads to the necessity to modify the basic Euler formula. It is obvious that each disconnected network of chains will obey the basic Euler formula. However, in such a complex, the outer boundary of each disconnected network must be properly related to some region of another graph. Either the network will be entirely internal to one polygon of the other graph (its chains can not cross those of the other graph) or it will share in the outer region of that other graph. The basic Euler formula is modified to take this into account. The modified Euler formula is stated as, "the total number of nodes and polygons must equal the total number of discrete components in the graph plus the number of chains in the graph plus one."

The modified Euler edit can be used on any planar graph. Although it does not ensure overall accuracy of the file, it is a good test of consistency among the counts of nodes, chains and polygons of the file. A failure of the modified Euler test is a guarantee that the file is not topologically consistent. However, passing this edit is no guarantee that the file is sound. Compensating errors can still exist that would allow the counts to balance.

CONCLUSION

This program is used in the exchange of digital map files with the U.S. Geological Survey. The polygonizing of the files is one of the first processes done to a file when it is received. After the Census Bureau performs the internal updates to the file, the TOPOLY program is run in edit mode. In this mode, it is one of the last processes run before returning the updated files to the U.S. Geological Survey. To date, the program has been run on over half of the 5Ø,ØØØ 7.5 minute quadrangles involved in building the National Digital Cartographic Data Base. The program has been converted successfully to run on the most recent TIGER file structure of the Census Bureau and it is anticipated to be run in the edit mode for years to come.

REFERENCES

Corbett, James P., 1979, Topological Principles of Cartography, Technical Paper 48, Bureau of the Census, Department of Commerce

Lefschetz, Solomon, 1975, Applications of Algebraic Topology , Springer-Verlag, New York

Nievergelt, A. and Preparata, F.P., 1982, "Plane-Sweep Algorithms for Intersecting Geometric Figures", Communications of the ACM, Vol 25 No 1Ø.

Prather, Ronald E., 1976, Discrete Mathematical Structures for Computer Science, Houghton Mifflin Company, Boston

White, Marvin, 1984, "Technical Requirements and Standards for a Multipurpose Geographic Data System", The American Cartographer, Vol 11 No 1.

"WYSIWYG" MAP DIGITIZING: REAL TIME GEOMETRIC CORRECTION AND TOPOLOGICAL ENCODING

Denis White and Jonathan Corson-Rikert
Laboratory for Computer Graphics and Spatial Analysis
Graduate School of Design, Harvard University
48 Quincy St., Cambridge, MA 02138
(617) 495-2526

Margaret Maizel
American Farmland Trust
1920 N St., Suite 400, N.W.
Washington, D.C. 20036
(202) 659-5170

ABSTRACT

Map input by manual digitizing no longer needs to be a multi-step process in which line gaps, overshoots, and topological coding errors are iteratively and painstakingly corrected. We have developed a "what you see is what you get" (WYSIWYG) approach to map digitizing that continuously displays on the computer screen a geometrically corrected and topologically structured representation of a map. This approach is analogous to the WYSIWYG style of word processing where insertions and deletions automatically cause lines, paragraphs, and pages to be adjusted such that a document is always displayed in its final form.

INTRODUCTION

Instruction in landscape architecture at Harvard includes courses and studios in regional scale landscape planning and design. For many years this instruction has included computer analysis of landscape suitability and environmental impacts. A major component of these projects is the building of the geographic data base. In the past, students have laboriously hand encoded data for soil types, vegetation types, elevation values, water features, cultural features, and other kinds of information into a grid cell format for use by the computer analysis programs.

In recent years the Laboratory for Computer Graphics and Spatial Analysis has assisted these courses and studios by developing software and related procedures for digitizing, verifying, and grid encoding large area data bases. These procedures and programs have been used in studies of Yosemite National Park, Minute Man National Historical Park, White Mountain National Forest, and Acadia National Park.

Our experiences have refined our thinking about a number of issues in map input to computers (Corson-Rikert and White, 1985a). With the support of the American Farmland Trust we are continuing to develop software to further improve this process. There are a number of key features of this software.

"WHAT YOU SEE IS WHAT YOU GET"

When our Lab's Odyssey system was designed ten or more years ago (White, 1979; Chrisman, 1979; Morehouse and Broekhuysen, 1982), there was virtually no moderately priced hardware that supported rapidly refreshed medium resolution interactive graphic display. A typical digitizing configuration was a digitizing table and a storage tube connected to a mainframe, or a storage tube micro standing alone with a digitizing table. In 1986, however, there is now no hardware limitation to immediately displaying points digitized, or even displaying points about to be digitized with so-called rubberbanding. CAD programs have been doing this for years; map digitizing programs should do likewise where appropriate.

With high speed refreshing, edited changes to a computer map data base can also be displayed immediately. Lines can be deleted, points can be inserted into a line or deleted from a line, nodal points where lines meet can be moved with their connected segments, and even entire lines can be rubberbanded into a translated position. Rapid display speed also allows arbitrarily positioned rectangular windows into the data base to be selected and displayed immediately.

GEOMETRIC CORRECTION

The speed of modern processors also allows for rubberbanding of coordinates registered to a map base with a global linear transformation. Piecewise linear rubbersheeting could certainly be accomodated as well for rubberbanding. Editing subsequent to initial map input from a digitizing table can often be done better with a mouse, but it should be easy to switch between mouse and table. Mouse editing normally occurs in map space without control point registration.

Geometric snapping of incoming points to existing points is another feature of the CAD environment quite suitable for map digitizing. In particular, this capability helps avoid the perennial plague of line overshoots and undershoots. Snapping in CAD is often enhanced by a grid of markers showing the points to which all input will be snapped. In map digitizing it is more appropriate to accept an input point exactly as digitized unless it falls within the tolerance range of a previously entered point, in which case it is merged with the older point. As a visual aid, points can be displayed with tolerance circles such that it is immediately clear whether a rubberbanded point will merge or not.

In fact, the snapping tolerance applies to a line as well. If an incoming point lies within the tolerance distance of a line segment (between the endpoints), an intersection should be formed and the old segment broken into two collinear segments. It is also possible, though potentially a strain on computational resources, to compute all intersections of a newly entered line with a number of older lines it crosses and display the new structure relatively quickly.

The rank ordering of the positional accuracy of features commonly found on maps is well supported in the computer data base by this method of map input. Features are entered in the order of most accurate to least accurate such that the latter are always snapped to the former when they are in close proximity.

TOPOLOGICAL ENCODING

Topological structuring of line and polygon networks on maps is now the standardly accepted method for insuring consistency and completeness in a computer map data base. The details of the dual incidence technique of encoding topology can be computed immediately and transparently while digitizing is taking place. In practice the construction of 2-cell topology in real time requires considerable extra work as new polygons are created and old ones destroyed by the entering of new lines. However, no additional steps need be required if polygons are to be labeled or tagged, since the 2-cell topology can be created during that process.

Successful implementation of instantaneous node topology maintenance is dependent on the correct operation of snapping within a tolerance and intersection finding. In this way, the need for an analysis of proximate points before creating final nodes of intersection is eliminated. (The overlay analysis of two or more polygon networks will require this analysis, however.)

FEATURE LABELING

The "spaghetti and meatballs" method of polygon labeling implemented in Odyssey was a major improvement over the method of labeling the left and right side polygons of each line. In interactive feature labeling, entering "meatballs" (or centroids or label points) is still useful since the points can be saved as text or symbol locations. A method requiring less input, however, is to highlight each feature in succession and prompt for its label (or tag). This process can be driven automatically, with the order in which features are presented determined by input order or perhaps more intelligently by a spatially sorted order. Either this semi-automatic naming method or the pick and name method can be used with point, line, or areal features.

CONTEXT

Often one layer or theme of a multi-layered map data base can be useful when digitizing another, particularly when features on separate layers partially or fully coincide. Displaying existing layers while entering a new one, another technique borrowed from CAD, is straightforward. Feature snapping or alignment from one layer to another is also desirable but less easy to implement.

When the eventual use of a computer map data base is for raster-based analysis (terrain analysis, viewsheds, path finding), visualizing the raster structure in the context of the vector data base helps determine the appropriate density of information to capture in vector form.

SEARCH OPTIMIZATION

Searching for the proximity of a new point in a very large data base can take a very long time unless some kind of spatial indexing or hierarchical structuring of the data base is used. One simple method is to limit searches to features within the current window. This method is consistent with the WYSIWYG philosophy; windows will tend to have a relatively constant density of information and be centered about the work at hand. Of course, computing the active list of features for a small window in a very large data base will be time consuming in itself.

IMPLEMENTATION

The digitizing strategies argued here have been implemented on three computers: initially and only partially in Pascal on an IBM PC/AT with a PGA graphics board, then translated to C on an Apple Macintosh, and finally in C on a Harris MCX/3 running Unix. The program, called Roots, is being used to create a data base for Clarke County, Virginia, under the sponsorship of the American Farmland Trust. The data base will include soils and topography (input of which will be commercially scanned), transportation features, hydrographic features, wells and sinkholes, and property boundaries and derivatives thereof such as zoning and farms. Most analysis of the data base will be done by the Geographical Resources Analysis and Support System (GRASS) developed by the Construction Engineering Research Laboratory of the U.S. Army Corps of Engineers.

CONCLUSION

For many years a major impediment to mapping and spatial analysis using computerized geographic data bases has been the exorbitant time and cost of entering map data into the computer. With the development of lower cost graphics hardware, integrated graphics operating systems on microcomputers, and improved algorithms for geometrical and topological processing of map data, this impediment is gradually being removed.

ACKNOWLEDGEMENTS

Portions of this work have been supported by the National Park Service and by IBM Corporation. Our work is continuing under the sponsorship of the American Farmland Trust, for whom we are helping to develop software and data bases to assist local government decision makers in natural resource conservation policies.

REFERENCES

Chrisman, N. 1979. "A Many Dimensional Projection of Odyssey", Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University.

Corson-Rikert, J. and White, D. 1986. "Comprehensive Map Input Software", *Proceedings, Urban and Regional Information Systems Association, 1986 Annual Conference.*

Corson-Rikert, J. and White, D. 1985a. "Issues in Map Digitizing", LCGSA Report 85-18, Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University.

Corson-Rikert, J. and White, D. 1985b. "The TRACE Program for Map Input", LCGSA Report 85-17, Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University.

Morehouse, S. and Broekhuysen, M. 1982. *Odyssey User's Manual,* Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University.

White, D. 1979. "Odyssey Design Structure", *Harvard Library of Computer Graphics, 1979 Mapping Collection,* Vol. 2, pp. 207-215.

SOFTCOPY METHODS OF CARTOGRAPHIC DATABASE MAINTENANCE

by
Lawrence L. Dambra, Scientist
Conse C. Vecchio, Principal Scientist
AUTOMETRIC, INCORPORATED
5205 Leesburg Pike
Suite 1308/Skyline One
Falls Church, VA 22041

ABSTRACT

Several automated techniques can be used in the cartographic change detection process to enhance productivity and accuracy, in support of geographic database maintenance. A series of experiments was performed to test the feasibility of using various change detection techniques in an automated cartographic production environment. Detected changes were flagged as feature updates to a geographic database. A hardware/software configuration testbed was constructed to simulate an automated production environment. Softcopy imagery and map/chart data were used to represent newly arrived source material. The geographic database was populated with cartographic feature vectors and attributes, henceforth referred to as Cartographic Feature Data (CFD). Production cartographers served as the experiment subjects to assure an operationally valid test sample. Results of the experiments are summarized on the following topics: Display Methods, Data Digitization, Image Manipulation, Zoom Factors, and Change Classification.

INTRODUCTION

Cartographic production agencies are rapidly incorporating softcopy technology to depart from the manual cartographic methods employed for years. An integral component of the softcopy movement is the digital cartographic feature database. The database is composed of geographically referenced and attributed feature data. It can be generated from a variety of sources including: maps/charts, imagery, reference graphics and textual sources. Feature data can be digitized and attributed to populate the database.

Once populated, changes to the digital database must be made as new source becomes available, in order to maintain the currency and accuracy of the database. The same types of sources used to populate the database can be exploited to maintain the geographic database. Raster scan digitizers provide a means to generate softcopy digital images of the hardcopy source material. The digital images can then be registered to a geographic frame of reference. Subsequently, the images may be displayed using methods that facilitate comparison with features in the geographic database. Anomalies can be identified and annotated in softcopy.

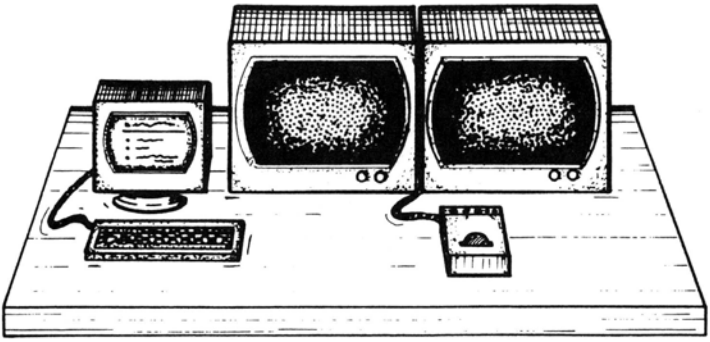
The softcopy concept described is the basis for the set of change detection experiments reported upon in this paper. The objective of the experiments was to test the feasibility of using various techniques to facilitate the change detection process in a softcopy environment. Analyst productivity and accuracy were also evaluated for given techniques. A description of the experiment design and implementation methodology and the experiment hardware/software configuration is presented. Each of the four change detection experiments are summarized to include: 1) a definition of the objectives; 2) the dependent and independent variables; 3) the experiment scenario; and 4) the results of the experiment.

EXPERIMENT DESIGN AND IMPLEMENTATION METHODOLOGY

Design of each of the four experiments began with determination of the independent variables to be manipulated and the dependent variables to be measured. A scenario was conceived which would support mensuration of data under the various states of manipulation. Using the scenario, data flow diagrams were developed to support the design activity and progress to implementation. A simple man-machine interface, which allowed option selection using hierarchical menus, was chosen for all of the experiments. This approach was selected to minimize: 1) the amount of time spent on implementation of the experiment software; 2) the amount of time required to orient the subjects; and 3) the influence of the man-machine interface on the outcome of the experiments.

EXPERIMENT WORKSTATION CONFIGURATION

The experiments were conducted on a workstation composed of the following hardware components: 1) VAX computer; 2) two high-resolution color image display monitors driven by a Gould IP-8500 image processor; 3) VT 220 alphanumeric CRT terminal; and 4) trackball graphic data entry and pointing device. The workstation is illustrated in Figure 1.



- ***ALPHANUMERIC MONITOR WITH KEYBOARD***
- ***TWO 1024 x 1024 IMAGE DISPLAY MONITORS***
- ***TRACKBALL BOX WITH FUNCTION KEYS***

Figure 1. The Experiment Workstation

Training Methodology

A Video Cassette Recorder (VCR) was used to train the experiment subjects regarding the objectives of the experiments. In addition, the hardware and software components of the experiment configuration were explained in the training video. The subjects were briefed on the specific tasks required of them for each experiment. The use of video tapes for training provided commonality between the experiment subjects in terms of introducing the experiments to each of the subjects. The video tape training was supplemented with hands-on training for each subject. The hands-on training allowed the subjects to work at the workstation with a training set that was developed for each experiment.

EXPERIMENT #1: IMAGERY CHANGE DETECTION TECHNIQUES

This experiment focused on determining how well an analyst could detect changes between vectors, (representing the features of the cartographic feature database), and a softcopy raster digital image display of monochrome imagery.

The independent variables for this experiment were: 1) display method; and 2) availability of a cartographic feature filtering function. The dependent measures for this experiment were: 1) speed of change detection; 2) accuracy of change detection; 3) use of a zoom/scroll function; 4) use of an image enhancement function; and 5) use of a cartographic feature filtering function.

The scenario for the experiment was as follows. Each analyst was exposed to four complete images and then corresponding features, extracted from a cartographic feature database. The images were segmented into patches, in one-quarter increments. The analyst viewed each of the four image patches and the corresponding feature data, using one of four display methods per image. Access to zoom, scroll, cartographic feature filtering, image enhancement functions was permitted at all times during the experiment. As the analyst detected changes between the database and the raster image, he/she used an electronic grease pencil function to annotate the change. After the analyst examined all four patches of a single image, a new image was displayed using a different display method. The process cycled until the analyst had examined all four images.

The display methods used were: 1) Split Screen; 2) Side-by-Side; 3) Overlay Superposition Method #1; and 4) Overlay Superposition Method #2. Each of the four display methods are presented in Figure 2. It is important to note that although the two overlay superposition methods appear very similar, Method #1 presented a reduced-resolution overview image on the left monitor, while Method #2 presented an overview line graphic. Both images appearing on the left monitor had a graphic monocle indicating the area of coverage displayed in full-resolution on the right monitor.

EXPERIMENT #2: MAP/CHART CHANGE DETECTION TECHNIQUES

This experiment was identical to Experiment #1 with the exception that the primary comparison source used was rasterized map/chart data. The use of rasterized map/chart source did not require the image enhancement capabilities, such as manipulation of the grayscale, which were provided in Experiment #1.

The objectives, dependent/independent variables, and the scenario were identical to Experiment #1. Reference the Experiment #1 description for details.

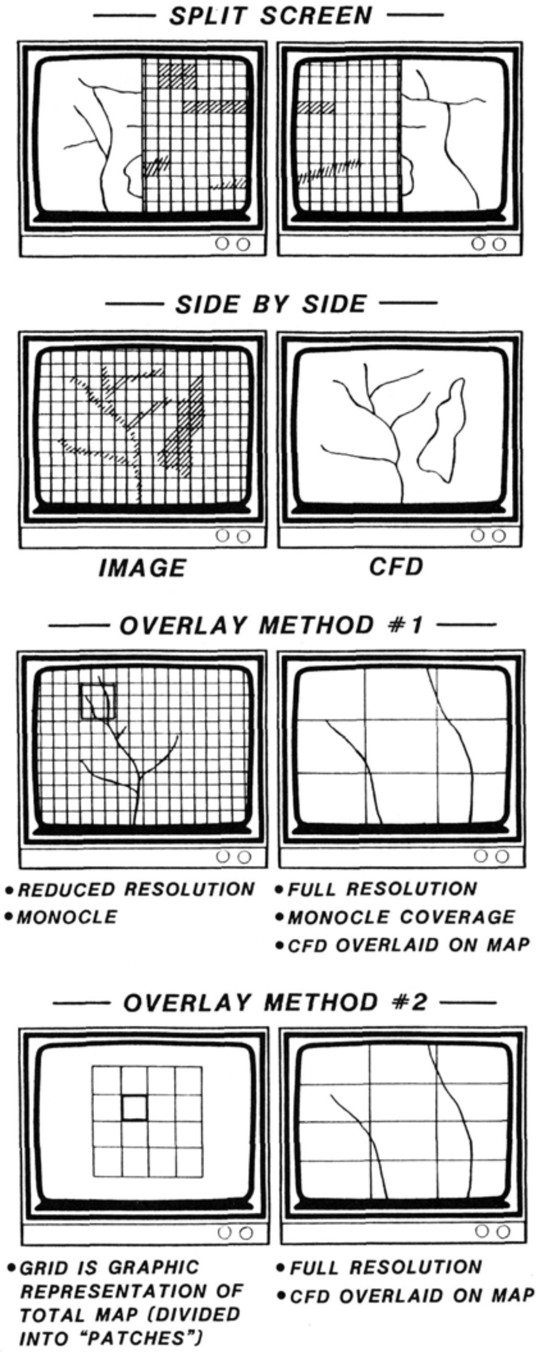


Figure 2. Display Techniques

EXPERIMENT #3: THE EFFECT OF RESOLUTION ON MAP/CHART CHANGE DETECTION

The purpose of this experiment was to evaluate the effect that varying the resolution of softcopy map/chart data has on the accuracy and productivity of softcopy change detection. Experiment #3 built upon the results of the first two experiments by taking advantage of the findings that indicated Overlay Superposition Method #1 to be the optimal display technique.

The dependent variables for this experiment were: 1) image resolution; 2) map/chart image number; and 3) experience of the cartographer. The independent measures for the experiment were: 1) speed of change detection; 2) accuracy of change detection; 3) use of the zoom/scroll function; and 4) use of the display toggle functions.

The scenario for the experiment was as follows. Three map sections were raster-scan digitized at each of three resolutions: 256 lines/inch, 384 lines/inch, and 512 lines/inch. These images were then displayed on the workstation using Overlay Superposition Method #1 along with the corresponding database features. Each subject viewed the three images at only one of the candidate resolutions. Viewing options to manipulate zoom/scroll, toggle various displays on/off, and filter the cartographic feature displays were available at all times. The analyst used the tools to perform change detection between the raster source and the database. The changes detected were marked as described in Experiments #1 and #2. The presentation combinations of resolution and image number were varied to normalize "learning curve" phenomena that would skew the results.

EXPERIMENT #4: CHANGE APPLICABILITY

The Change Applicability experiment was designed to determine how identified feature changes impact the database from the perspective of product generation. Given the case where several products are produced from a single database, a change may not impact all of the products produced. Obviously, product scale is a major factor regarding applicability of change to a product. If a change can be codified to a fine level of attribution, a generic feature-to-product content look-up table can be created which determines product applicability of a change.

The independent variables manipulated in this experiment were: 1) source type (imagery or map/chart); 2) method of codification (automated or manual); and 3) level of subject cartographer's experience. The dependent measures were: 1) speed of change codification or applicability assessment; 2) accuracy of change codification or applicability assessment; and 3) use of image display manipulation tools (e.g., zoom/scroll, feature filter, and image enhancement).

The scenario was as follows: The experiment subjects were presented a mix of softcopy images that are map/chart and imagery based. Feature changes on the image were annotated with Minimum Bounding Rectangles (MBRs). The experiment was designed to resume where the other experiments ended. That is, changes had already been discovered and automated. Now the subject must determine the nature of the change and the impact that change has on a given set of cartographic products. Two separate groups were established to test two distinct techniques. The first group categorized the change and determined applicability aided by softcopy product specifications. The second group categorized the change using a generic attribute coding system that forced the subject to classify the change into a feature type. A look-up table was constructed that mapped feature types to products. Therefore, once the change was classified, the applicability to the given set of products was determined automatically by invoking the look-up table. The look-up table was constructed by extracting product-specification data and incorporating that data as the relation criteria.

EXPERIMENT RESULTS

The experiment results were based on the following:

a. Statistical analysis of subject performance;
b. Subject preference data from questionnaires;
c. Experiment proctor observations.

Display Methods

The variance of performance noted for the display methods tested proved to be insignificant. That is, the variance for speed and accuracy between the four display methods was minimal. The raw scores for the Side-by-Side display method ranked slightly higher than the others; however the difference was less than the computed standard deviation. Given the small sample size (12 subjects for CD1 and CD2) the insignificance of variance was not a surprise.

Thus, the recommendation to provide more than one display method is supported on the basis of analyst preference versus statistical results. Based on the data extracted from the experiment questionnaires and proctor observations, the following conclusion was formulated: "The individuality of each analyst is a significant factor in determining the most favored display method". For example, although the Overlay Method #1 proved to be the most preferred, a subset of analysts preferred the Side-by-Side method. It appeared that the optimum method of display was highly situation-dependent. Factors such as feature density, type of feature, and characteristics of the geographic area in which the change occurred, had a significant effect on the analysts' ability to discriminate changes. Therefore, it is recommended that more than one display method be provided to support softcopy change detection in a production system. This would provide flexibility and enhance user acceptance of a softcopy system.

Data Digitization

The 256 LPI resolution is the recommended resolution based on the experiment results. The Analysis of Variance (ANOVA) for total patch time provided the mean time expended by the analysts for each map/chart patch. The mean time for each resolution was calculated. The mean average time per image was calculated by multiplying the 384 and 512 patch times by four (4) (there were four patches per image). The mean average image times for each resolution were as follows:

- 256 LPI: 24.2 minutes
- 384 LPI: 37.3 minutes
- 512 LPI: 40.1 minutes

As expected, the time expended per patch increased as the resolution of the digital map/chart data increased. The analysts were required to review four patches for the 384 and 512 LPI images. The 256 LPI image contained only one patch. Although the 512 resolution patches covered less geographic area than the lower resolutions, analysts did not spend a proportionately lesser amount of time on these patches. Each patch, regardless of resolution, was treated as an individual image, thus the total image time for the 512 resolution was largest.

As a result of time spent on each patch, the number of errors of commission and omission increased as the resolution increased. The number of changes not graded also increased; this was undoubtedly due to CFD misalignment which was more apparent at the higher resolutions.

Analyst preferences for the three resolutions were documented in the experiment questionnaires. The percentages of preference are as follows:

- 256 LPI: 4%
- 384 LPI: 44%
- 512 LPI: 52%

The majority of the analysts preferred the higher resolutions to support the requirements of the experiment. However, the timeline and accuracy data collected support the use of the lower resolution 256 LPI for most products (to the 1:50,000 scale). Higher resolutions would be recommended for 1:24,000 scale products and smaller.

Image Manipulation

The experiment analysts were provided the following toggle capabilities in the experiment:

- CFD Toggle (toggle vectoried CFD)
- Change Annotation Toggle (toggle MBRs)
- Map/Chart Base Toggle (toggle rasterized source)

Change Detection Experiments #1 and #2 tested the feasibility of using CFD toggle. It proved to be a valuable capability and is recommended for a production workstation. The subjects used CFD toggle as their primary change detection technique. Results of Change Detection Experiment #3 were consistent with Experiments #1 and #2 for this option.

The experiment subjects used the CFD toggle capability to create a flicker effect by holding the toggle key down on the keyboard. The flicker effect of CFD over the rasterized base made it easier to compare the CFD with the base. The analysts used the CFD toggle ten times more than the other toggles. It was also noted that the experienced analysts used the CFD toggle much more than the inexperienced analysts. The results of this experiment reinforce the need to provide the CFD toggle capability in a production environment. The other toggles tested should be evaluated to determine the cost impact of providing these capabilities, weighed against the added enhancement of workstation tools.

Zoom Factors

The zoom and scroll factor capabilities were used extensively in the experiments. The use of the zoom capability was inversely proportionate to the resolution. That is, analysts examining the 256 LPI resolution used the zoom capability twice as much as the analysts who viewed 512 LPI resolution patches. In addition, analysts viewing the 384 LPI resolution patch used the zoom capability approximately 1 1/2 times as much as those viewing the 512 LPI resolution patches (see Figure 3). This suggests that the analysts used the zoom capability to create a similar field-of-view image for all resolutions.

Eight zoom factors were provided to the experiment analysts for each resolution. It is noted that zoom factors one through five were used extensively during the experiments. The use of zoom factors six through eight was substantially less. This is due primarily to the fact that the quality of the rasterized graphic diminished at the higher zoom factors. The zoom function was implemented in hardware as a simple pixel replication. At zoom factors above five, a very strong aliasing effect occurred.

Given the heavy use of zoom capability, it is recommended that the capabilities be provided in a production environment. This recommendation is supported by the statistical analysis of subject performance and positive preference responses by the analysts.

Change Classification

The generic code assignment and look-up table based technique is the recommended technique for determining change applicability. The generic codes technique requires the analyst to classify the feature change and tag the change with the appropriate code. After the changes are coded, they are compared to a Change Applicability Matrix (CAM) that maps generic codes to applicable products.

The manual method tested required the analysts to review product-specification help files to aid in the determination of change applicability. In the experiment, a subset of five products was established. Timeline results of experiment subjects did not differ significantly. However, if the set of products was greater than five the analysts would have spent significantly more time using the product-specification method when contrasted with the generic code based method.

Accuracy, when contrasted with codification method, is the other important factor for change applicability. Change applicability cannot be performed accurately unless the changed feature is classified correctly. The experiment results indicated that the analysts which used the generic code method scored slightly higher than those who used the product-specification method.

In the product-specification method there are two possible sources of error. The first is the identification of the feature, and the second is the review of product-specifications to determine applicability. For the generic code method, the only source of error is changed feature identification. This assumes that the CAM can be validated to assure correct applicability.

The results of this experiment support the use of a generic coding method in terms of time and accuracy. In addition, the results emphasize the need to provide the tools necessary to assure correct feature identification.

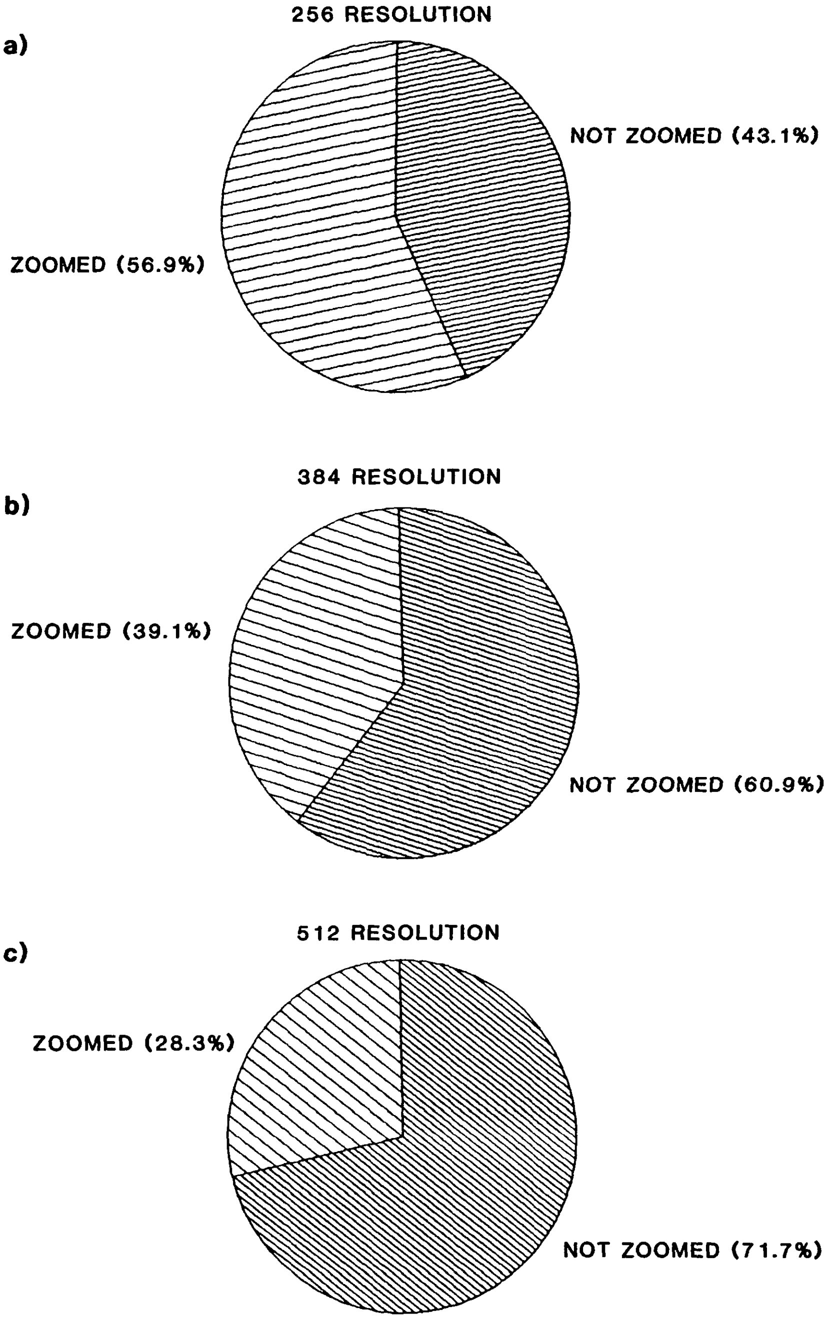


Figure 3. Zoom Usage

TESTING A PROTOTYPE SPATIAL DATA EXCHANGE FORMAT-- THE FEDERAL GEOGRAPHIC EXCHANGE FORMAT EXPERIENCE

Robin G. Fegeas
U.S. Geological Survey
521 National Center
Reston, Virginia 22092

ABSTRACT

Successful exchange of digital cartographic and geographic data is dependent upon many factors. Recent standards development activities have attempted to address some of these factors, including data exchange formats. The Standards Working Group of the Federal Interagency Coordinating Committee on Digital Cartography has developed a prototype spatial data exchange format. This format, the Federal Geographic Exchange Format (FGEF), is designed to be a Federal governmentwide standard for the exchange of digital cartographic data, geographic data, spatially referenced data and associated attribute data. To help determine its strengths and weaknesses, assess its feasibility for use, and provide data for refinements and improvements, the format is being tested by a number of Federal agencies. Four levels of testing have been identified: (1) an agency tests the format for its ability to handle just its own data, (2) an agency develops capabilities to handle all FGEF data types, (3) two agencies exchange selected data types, and (4) two agencies exchange files that test all data types. The testing of the prototype FGEF is expected to be completed in the spring of 1987. This paper outlines the methodology used and reports on some preliminary testing results.

Publication authorized by the Director, U.S. Geological Survey.

Integrating Multiple Data Representations For Spatial Databases

David M. McKeown, Jr.
Robert Chi Tau Lai
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA. 15213

Abstract

An intelligent spatial database must be able to organize and store information from diverse sources. Aerial imagery, map, and terrain data must be merged with textual and collateral information. Future systems will integrate the results of automated analysis of remotely sensed imagery within the context of the spatial database. No single internal representation can efficiently provide for the variety of the needs and problems for these types of spatial databases. For example, in order to efficiently search large databases it is critical to be able to partition the search based on spatial decompositions, whether hierarchical, regular, or mixed. In this paper we discuss some recent work on integrating multiple spatial and factual data representations so as to capitalize on their inherent advantages for search, geometric computation, and maintenence of topological consistency.

1. Introduction

An intelligent spatial database must be able to organize and store information from diverse sources. Aerial imagery, map, and terrain data must be merged with textual and collateral information. Future systems will integrate the results of automated analysis of remotely sensed imagery within the context of the spatial database. No single internal representation can efficiently provide for the variety of the needs and problems for these types of spatial databases. For example, in order to efficiently search large databases it is critical to be able to partition the search based on spatial decompositions, whether hierarchical, regular, or mixed. However, the data structures used for such a decomposition, hierarchy trees, quadtrees, and k-d trees, are not particularly well suited to (for example) the maintenance of topological consistency. Arc-node, or segment-node representations have been developed for this purpose, but they introduce problems for spatial decomposition algorithms. Finally, neither addresses the problem of feature attribution, coupling semantic descriptions of the feature with its spatial component. Semantic networks or frame-based systems can be expected to compete with relational models in this area.

Thus, there are several dimensions along which one can choose appropriate data structures and representations. In this paper we describe some research results in integrating multiple data representations within the context of an experimental spatial database system, MAPS. developed at Carnegie Mellon University. The areas covered are:

- The use of a schema-based description that allows queries based on user-defined attributes as well as shape, size, and spatial relationships computed and maintained by the system.
- The maintenance of an arc-node feature representation for feature editing and display while maintaining a parallel entity-based spatial database in a hierarchical containment tree.
- Some comparisons of the relative properties and merits of various component databases for storage and retrieval of data in different representations.

2. An Overview of MAPS

The MAPS spatial database[1, 2, 3, 4] was developed between 1980-1984 supported by the DARPA Image Understanding Program as research into large-scale spatial databases and spatial knowledge representation. It is interesting that this system has expanded from its original research goal of developing an interactive database for answering spatial queries into a component of several knowledge-based image understanding systems under development at Carnegie Mellon University. MAPS is a large-scale image/map database system for the Washington D.C. area that contains approximately 200 high resolution aerial images, a digital terrain database, and a variety of map databases from the Defense Mapping Agency (DMA). MAPS has been used as a component for an automated road finder/follower, a stereo verification module, and a knowledge-based system for interpreting airport scenes in aerial imager. In addition, MAPS has an interactive user query component that allows users to perform spatial queries using high resolution display of aerial imagery as an method for indexing into the spatial database. This capability to relate, over time, imagery at a variety of spatial resolutions to a spatial database forms a basis for a large variety of interpretation and analysis tasks such as change detection, model-based interpretation, and report generation.

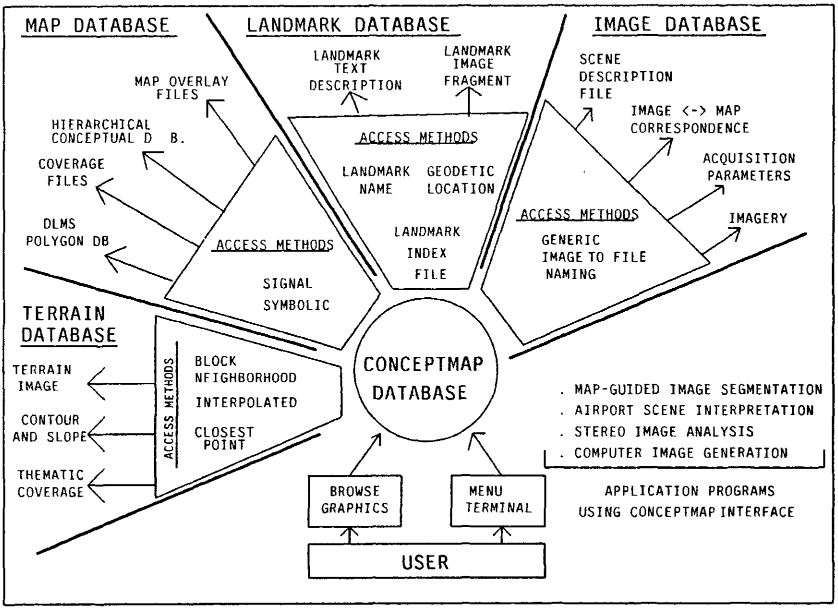


Figure 2-1: MAPS: System Overview

Figure 2-1 shows the system organization of MAPS. Four databases are maintained within MAPS: a digital terrain database, a map database, a landmark database, and an image database. A fifth database, CONCEPTMAP, consists of a schema-based representation for spatial entities and a set of procedural methods that provide a uniform interface to each of the four component databases for interactive users or application programs. It is this interface that allows us to represent and access image, map, terrain, and collateral data in a manner that best suits the intrinsic structure of the data. At the same time the CONCEPTMAP database provides uniform access to a variety of spatial data independent of the particular internal structure. This is in sharp contrast to methods proposed for uniform representation of image and cultural data such as raster data sets and regular decompositions such as quadtrees or k-d trees. In the following sections we touch on some interesting aspects of the CONCEPTMAP database. Figure 2-2 gives another view of the structure of spatial data within the MAPS system, that of the physical representation of data as stored in the component databases. There are currently four data representation types: semantic knowledge, geometrical and topological descriptions of spatial data, raster representation, and spatial hierarchies.

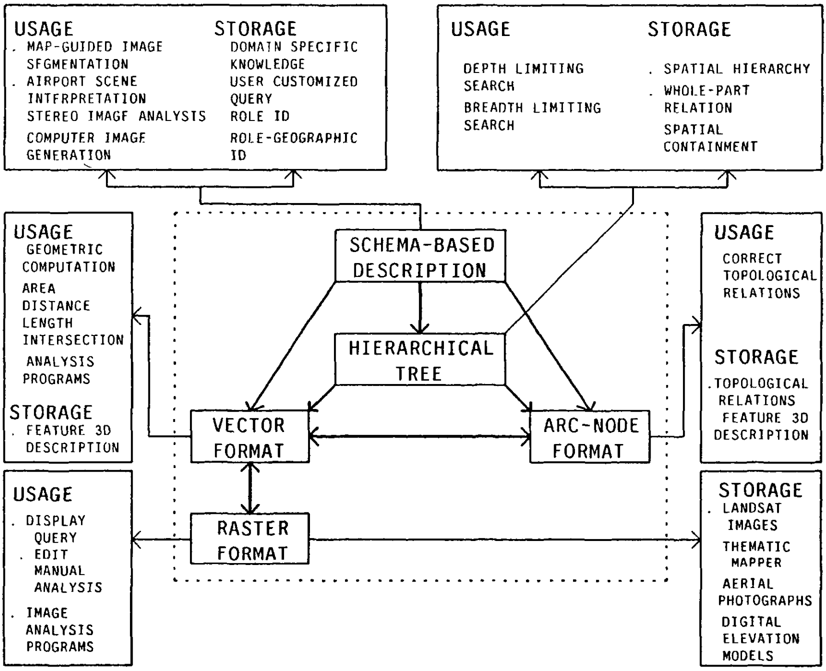


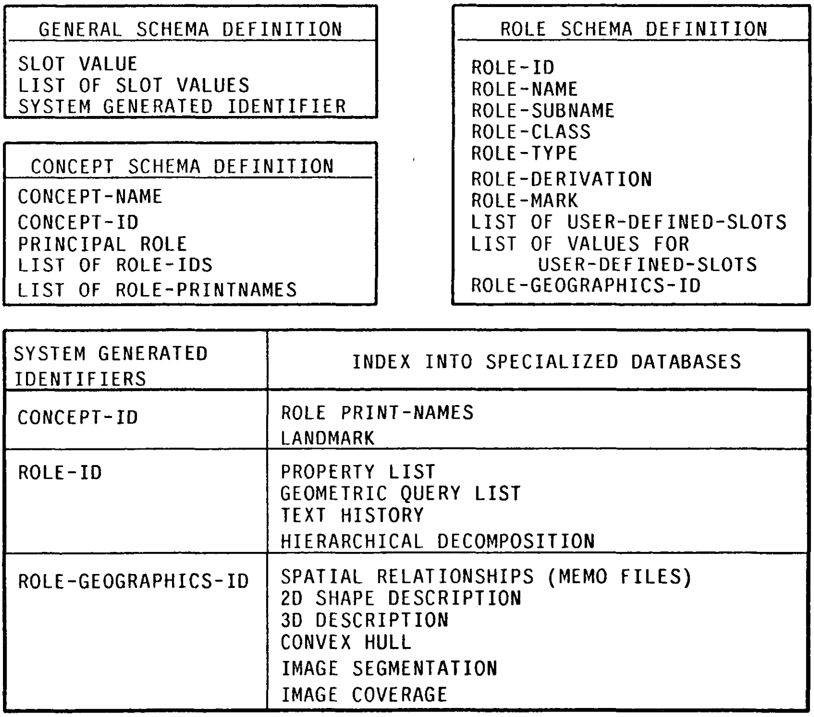
Figure 2-2: Data Representations In MAPS

A key point is that there is not a one-to-one mapping between the source or type of data and its representation methods within the MAPS system. For example, raster formats are used to store both digital image data as well as digital elevation models since this is the natural representation even though the access semantics for a two-dimensional image are different than for a three-dimensional DEM. The access functions associated with each datatype implicitely make use of information concerning properties of the raster such as sensor and camera models for image access and elevation cell size and ground coordinate when accessing elevation data. In the case of map features their coordinates can be stored as in either vector or arc-node formats, and relationships between features can be conputed in either representation. This flexability allows MAPS to provide flexible access to spatial entities using several independent methods. Thus, the location of a spatial entity can be retrieved via its intrinsic properties, as stored in the schema-based description, its relationships with other entities via a hierarchical containment tree, or via topological relations maintained within an arc-node representation.

As shown in Figure 2-2 there are many relationships between data that are explicitely stored within each representation. For example, the hierarchical containment tree is generated using the CONCEPTMAP schemata and their associated spatial data as stored in vector format. Once generated, the containment tree can be used to efficiently search our arc-node representation for features within an arbitrary area of interest using spatial decomposition based on the underlying structure of the area. It is the role of the CONCEPTMAP interface to support conversions between different representations and spatial data retrieval in a manner that hides the actual physical representation(s) of the underlying data. Given this organization application programs can be written which capitalize on the most efficient data access methods, and can use these representations as primitives to construct customized access and query mechanisms to support specific tasks. In the following Section we discuss the organization of the CONCEPTMAP schema-based represention. This representation stores the semantics of each spatial entity as well as symbolic methods for access of the associated spatial data. In Section 4 we briefly discuss some experiments in representation of spatial data using vector and arc-node representations.

3. A Schema-BasedRepresentationFor Spatial Entities

The CONCEPTMAP database uses a *schema-based* representation for spatial entities. Using schemas (or frames) is a well understood AI methodology for representing knowledge. Such a representation can be combined within several problem-solving methods such as semantic networks, scripts or production systems to construct a problem-solving system[5]. Each entity in the CONCEPTMAP database is represented by one *concept* schema and at least one *role* schema. A *concept* can represent any spatial object and associates a name with a set of attributes stored in the *concept* and *role* schemata. Figure 3-1 gives definitions of the slot names for *concept* and *role* schemata. Figure 3-2 gives an partial list of the concepts in the MAPS WASHDC database.



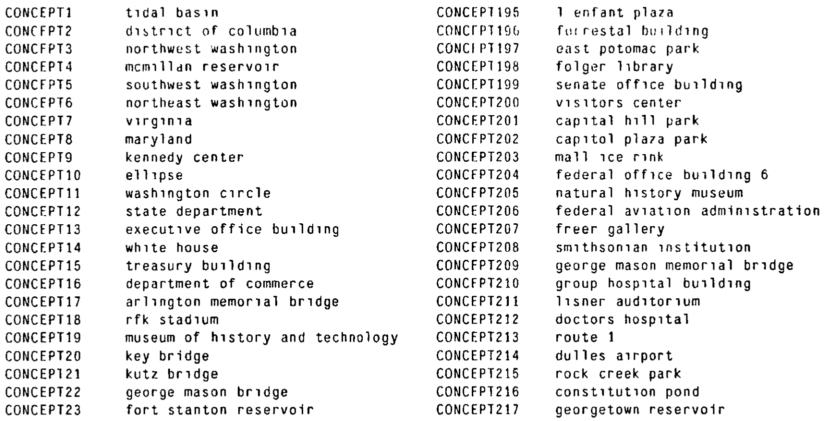
GENERAL SCHEMA DEFINITION
SLOT VALUE LIST OF SLOT VALUES SYSTEM GENERATED IDENTIFIER

CONCEPT SCHEMA DEFINITION
CONCEPT-NAME CONCEPT-ID PRINCIPAL ROLE LIST OF ROLE-IDS LIST OF ROLE-PRINTNAMES

ROLE SCHEMA DEFINITION
ROLE-ID ROLE-NAME ROLE-SUBNAME ROLE-CLASS ROLE-TYPE ROLE-DERIVATION ROLE-MARK LIST OF USER-DEFINED-SLOTS LIST OF VALUES FOR USER-DEFINED-SLOTS ROLE-GEOGRAPHICS-ID

SYSTEM GENERATED IDENTIFIERS	INDEX INTO SPECIALIZED DATABASES
CONCEPT-ID	ROLE PRINT-NAMES LANDMARK
ROLE-ID	PROPERTY LIST GEOMETRIC QUERY LIST TEXT HISTORY HIERARCHICAL DECOMPOSITION
ROLE-GEOGRAPHICS-ID	SPATIAL RELATIONSHIPS (MEMO FILES) 2D SHAPE DESCRIPTION 3D DESCRIPTION CONVEX HULL IMAGE SEGMENTATION IMAGE COVERAGE

Figure 3-1: MAPS: Concept and Role Schemata Definitions

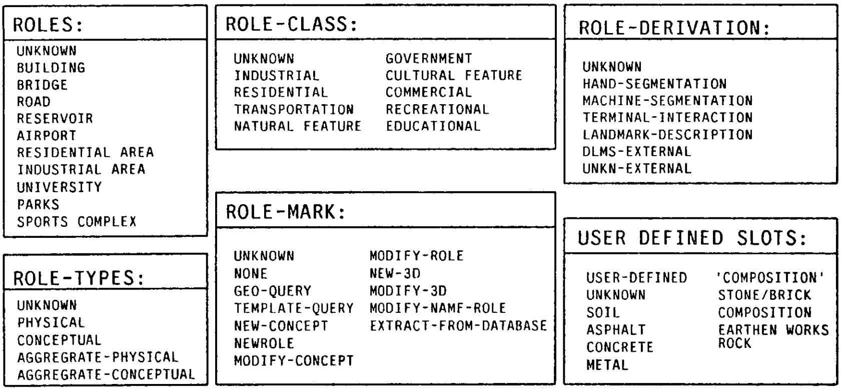


```
CONCEPT1     tidal basin                        CONCEPT195   1 enfant plaza
CONCEPT2     district of columbia               CONCEPT196   forrestal building
CONCEPT3     northwest washington               CONCEPT197   east potomac park
CONCEPT4     mcmillan reservoir                 CONCEPT198   folger library
CONCEPT5     southwest washington               CONCEPT199   senate office building
CONCEPT6     northeast washington               CONCEPT200   visitors center
CONCEPT7     virginia                           CONCEPT201   capital hill park
CONCEPT8     maryland                           CONCEPT202   capitol plaza park
CONCEPT9     kennedy center                     CONCEPT203   mall ice rink
CONCEPT10    ellipse                            CONCEPT204   federal office building 6
CONCEPT11    washington circle                  CONCEPT205   natural history museum
CONCEPT12    state department                   CONCEPT206   federal aviation administration
CONCEPT13    executive office building          CONCEPT207   freer gallery
CONCEPT14    white house                        CONCEPT208   smithsonian institution
CONCEPT15    treasury building                  CONCEPT209   george mason memorial bridge
CONCEPT16    department of commerce             CONCEPT210   group hospital building
CONCEPT17    arlington memorial bridge          CONCEPT211   lisner auditorium
CONCEPT18    rfk stadium                        CONCEPT212   doctors hospital
CONCEPT19    museum of history and technology   CONCEPT213   route 1
CONCEPT20    key bridge                         CONCEPT214   dulles airport
CONCEPT21    kutz bridge                        CONCEPT215   rock creek park
CONCEPT22    george mason bridge                CONCEPT216   constitution pond
CONCEPT23    fort stanton reservoir             CONCEPT217   georgetown reservoir
```

Figure 3-2: Concepts from 'washdc' CONCEPTMAP Database [partial list]

There are three unique identifiers generated by the CONCEPTMAP system which allow for indirect access to additional factual properties of concept or role schemata.

- The *concept-id* is unique across all concepts in all CONCEPTMAP databases. That is, given a concept-id one can uniquely determine the name of the spatial entity.
- The *role-id* uniquely determines a role schema across all CONCEPTMAP databases.
- The *role-geographics-id* uniquely determines a collection of points, lines or polygons in vector notation. Each point is represented as <latitude,longitude,elevation>.



ROLES:
UNKNOWN
BUILDING
BRIDGE
ROAD
RESERVOIR
AIRPORT
RESIDENTIAL AREA
INDUSTRIAL AREA
UNIVERSITY
PARKS
SPORTS COMPLEX

ROLE-TYPES:
UNKNOWN
PHYSICAL
CONCEPTUAL
AGGREGRATE-PHYSICAL
AGGREGRATE-CONCEPTUAL

ROLE-CLASS:
UNKNOWN
INDUSTRIAL
RESIDENTIAL
TRANSPORTATION
NATURAL FEATURE
GOVERNMENT
CULTURAL FEATURE
COMMERCIAL
RECREATIONAL
EDUCATIONAL

ROLE-MARK:
UNKNOWN
NONE
GEO-QUERY
TEMPLATE-QUERY
NEW-CONCEPT
NEWROLE
MODIFY-CONCEPT
MODIFY-ROLE
NEW-3D
MODIFY-3D
MODIFY-NAMF-ROLE
EXTRACT-FROM-DATABASE

ROLE-DERIVATION:
UNKNOWN
HAND-SEGMENTATION
MACHINE-SEGMENTATION
TERMINAL-INTERACTION
LANDMARK-DESCRIPTION
DLMS-EXTERNAL
UNKN-EXTERNAL

USER DEFINED SLOTS:
USER-DEFINED
UNKNOWN
SOIL
ASPHALT
CONCRETE
METAL
'COMPOSITION'
STONE/BRICK
COMPOSITION
EARTHEN WORKS
ROCK

Figure 3-3: Conceptmap Database Dictionary:
System and User Defined Attributes

As shown is Figure 3-1 these identifiers are also used to index into other components of the MAPS database. For example, the *concept-id* is used to search for landmark descriptions of measured ground control points used during the calculation of transform functions for image-to-map and map-to-image

ROLE: BUILDING	
UNKNOWN	MEDICAL CENTER
OFFICE BUILDING	BOATHOUSE
GOVERNMENT BUILDING	APARTMENTS
DORMITORY	HOTEL/MOTEL
CONCERT HALL	LIGHT MANUFACTURING
MUSEUM	GYMNASIUM
PERFORMING ARTS COMPLEX	HOSPITAL
RAILROAD STATION	LIBRARY
ADMINISTRATION	CLASSROOMS
MEMORIAL	STUDENT UNION

ROLE: AIRPORT
UNKNOWN
COMMERCIAL
MILITARY

ROLE: RESERVOIR
UNKNOWN
DRINKING WATER

ROLE: UNIVERSITY
UNKNOWN
DORMITORIES
ATHLETIC FACILITIES
UNIVERSITY CAMPUS
RESEARCH FACILITY
ADMINSTRATION
STUDENT CENTER
CAFETERIA
ADMISSION

ROLE: ROAD	
UNKNOWN	TRAFFIC CIRCLE
INTERSTATE HIGHWAY	INTERSECTION
STREET	INTRACITY HIGHWAY
AVENUE	ACCESS ROAD
RURAL ROAD	

ROLE: BRIDGE
UNKNOWN
RAILROAD
PEDESTRIAN
AUTOMOBILE

ROLE: SPORTS COMPLEX	
UNKNOWN	OPEN AREA
STADIUM	GOLF COURSE
BOAT MARINA	
ICE SKATING RINK	

ROLE: PARKS	
UNKNOWN	ZOO
PLAYING FIELD	FORMAL GARDEN
OPEN AREA	BLEACHERS
POND	SCULPTURE GARDEN
FORESTED AREA	

ROLE: POLITICAL
UNKNOWN
STATE
COUNTY
CITY
DISTRICT

ROLE: RESIDENTIAL AREA
UNKNOWN
SINGLE FAMILY HOUSING
APARTMENT COMPLEX
MIXED HOUSING

Figure 3-4: Conceptmap Database Dictionary: Subrole Attributes

correspondence. The *role-id* is used as the basic entity when building a hierarchy tree decomposition. The *role-geographics-id* is used to acquire the unique geographic position for a *role schema* as well as for linkage into the MAPS image database and segmentation files generated by human interaction or machine segmentation. There are three reasons for this approach. First, it allows CONCEPTMAP to handle very large databases with a minimal amount of information resident in the application process. The identifiers provide a level of indirection to the actual data, which is stored in a variety of formats and may or may not be present for a large subset of the database. Second, we can achieve a great deal of flexibility and modularity in processes which communicate about spatial entities. Given the name of a CONCEPTMAP database, a *concept-id* or *role-id* uniquely determines the entity in question. This facilitates the construction of application programs with simple query structures, requiring a minimum of communication overhead. Finally, given this decoupling from the CONCEPTMAP database, each of the MAPS component databases, image database, terrain database, landmark database, and map database may be physically resident on a different workstation or mainframe.

There are three levels of attribution available to users within CONCEPTMAP:

• *system-wide* attributes:	stored in role schema.
• *user-defined* attributes:	stored in role schema.
• *property-list* attributes:	stored in property list database.

CONCEPTMAP allows users to define additional attributes, called *user-defined*, similar in function to the *role-name* and *role-subname* slots described above. Finally, *property-list* attributes can also be defined by the user and are capable of representing a variety of datatypes including 'strings', 'integers', 'double',and 'list' using a simple data structure based on lists of the following:

```
<'attribute-name' , 'attribute-value'>
```

Attributes of all three classes are interpreted by CONCEPTMAP using a database dictionary defined for each class type. CONCEPTMAP can be easily configured for a particular application such as geology or forestry simply by developing an appropriate database dictionary. *User-defined* and *property-list* attributes can be defined dynamically by a user at an interactive session. Figure 3-3 gives a partial dictionary of the *system-wide* slots and representative values for a CONCEPTMAP database. Figure 3-4 is a partial dictionary of *role-subname* values associated with *role-name* values in Figure 3-3. A more complete description of the schema structure for the CONCEPTMAP database, and the generation of hierarchical containment trees and their use in spatial search can be found in[4].

4. Mixed Representationsfor Spatial Features

In this section we expand upon our description of Figure 2-2. We discuss the use of vector formats to represent individual spatial entities within the MAPS system, the use of arc-node structures to maintain topological consistency among collections of entities organized as a CONCEPTMAP database, and the organization of spatial entities into a hierarchical containment tree for efficient spatial search. Finally we discuss some performance and sizing results in the context of two CONCEPTMAP databases.

4.1. Entity Based Vector Format

As described in Section 3 each entity in a CONCEPTMAP database is represented by one *concept* schema, and at least one *role* schema. Each *role* schema can define a point, line, or polygon represented by collections of *<latitude,longitude,elevation>* triples and given a system-wide unique identifier, *role-geographic-id*. The use of vector format on a per entity basis allows for simple per feature geometric tests and the incremental (independent) accumulation of spatial entities from a variety of sources. For example it is relatively easy to automatically convert external external databases to vector format or to allow for human delineation using graphics overlay and recovery of geographic position via image-to-map correspondence. Other issues such as the desire to partition large databases over multiple workstations raises the possibility of spatial entities being represented in several databases simultaneously. Further, hierarchical descriptions are created within the context of a particular database on a per entity basis. Thus, a representation for spatial data which treats each entity with maximal independence satisfies many of these requirements. However, it should be obvious that this independence assumption raises issues in the maintenence of topological consistency especially for entities with shared boundaries, inconsistencies that arise from errors in image-to-map correspondence and scale and accuracy mismatches. In the following section we briefly describe our attempts to reconsile these issues.

4.2. Topological Consistency Among Spatial Entities

While the *role-geographic-id* is used within a CONCEPTPMAP database to uniqely define a spatial entity, we have extended its use as a method to represent all features within a particular CONCEPTMAP database in arc-node format. The arc-node format used in MAPS is the Standard Linear Format, SLF, which has been defined and studied by Defense Map Agency (DMA) for possible use as an internal digital data exchange format. The version of SLF that is currently implemented within MAPS uses the DMA Feature File, DMAFF, to represent limited feature semantics on a per feature basis. One can view DMAFF as a dictionary of legal cartographic features and a set of attributes used to describe properties of those features. As is well known, arc-node and other related formats explicitely represent the topological characteristics of collections of features in terms of shared boundaries, points of intersection, and some limited ability to represent containment and holes. These topological relationships can be retrieved without further computation, but for complex databases may require tradeoffs between large internal working sets and linear search.

In order to use arc-node format to maintain topological consistency we must be able to convert collections of vector format CONCEPTMAP entities into an arc-node representation. We make use of geometric information already stored in the CONCEPTMAP database such as points of intersection and common coordinate points to create *segments* and *nodes*. *Features* are defined in terms of *segments* and the segment direction within each feature and are maintained consistent with their CONCEPTMAP database counterparts by use of the *concept-id*, *role-id*, and *role-geographics-id*. Since many of the necessary spatial relationships are already computed in vector format and are stored in memo files in CONCEPTMAP the process is primarily one of generating of new nodes and segments based upon points of intersection between features. Thus, each *role-geographic-id* will generate one feature in arc-node format. In the case of partial overlap or ambiguous mismatches an arc-node editor is manually used to correct the topological relationships among the features. This interaction may cause the actual feature coodinates to be updated as in adjustment for slivers and gaps between adjacent features sharing a common boundary. Once the spatial data is converted and inconsistencies are removed we can regenerate the *role-geographic-id* database by traversing and enumerating the arc-node database on a per feature basis. Thus arc-node format is used to maintain topologically consistent collections of features that are stored and manipulated outside of the arc-node representation as independent entities.

An interesting extension to the use of arc-node format to maintain topologically consistent collections of features is in the assimilation of external databases stored in SLF format into the CONCEPTMAP database representation. Figure 4-1 shows the process by which the DMAFF attribute data is used to generate slot values for CONCEPTMAP role schema, while the topological data is used to generate the corresponding spatial entities. The DMAFF attribute sets associated with each feature are automatically translated into *concept* and *role* schemata or property lists in CONCEPTMAP. As before coordinate conversion from arc-node to vector format is accomplished by expansion of the *feature-segment-node* representaion to the vector point list.

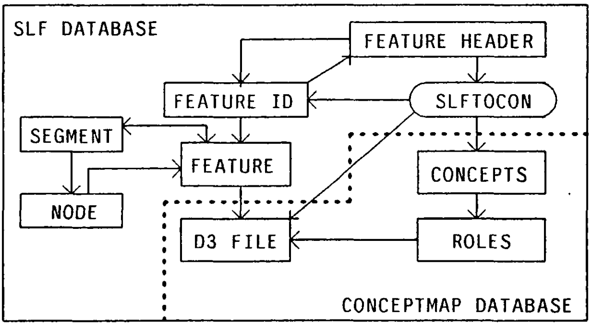


Figure 4-1: Converting SLF To CONCEPTMAP Database

4.3. Measuring Database Complexity For Vector and Arc-Node Data

In this section we briefly describe an experiment to gather empirical data on storage requirements and representation complexity for spatial data stored as vectors on a per feature basis and as topologically consistent arc-node collections. We took two CONCEPTMAP databases, WASHDC and USA, having very different properties and investigated their representation in vector and arc-node formats. The WASHDC database was composed of over 300 spatial entities in the Washington D.C. area. It consists of features such as buildings, roads, bridges, neighborhoods, and political boundaries. The USA database consists of the boundaries of the 50 states in the U.S.. In some sense these databases are at extremes in terms of their topological and geometic properties. The WASHDC database contains large numbers of isolated features such as buildings and neighborhoods, large numbers of features with sparse intersections such as roads, and relatively small numbers of features with shared boundaries, primarily political and large natural features such as rivers. Most of the features were either lines or polygons with small number of vector points. The USA database consisted of polygons with large numbers of vectors points and many shared boundaries. Figure 4-2 shows some statistics for both databases in terms of number of segments, nodes, and points per feature.

	number of features	number of segments	number of nodes	number of points	points /feature	points /node	points /seg	nodes /seg	nodes /fea	seg /fea
SOUTH WEST										
vector	4			2775	893.75					
converted	4	69	59	2767	891.75	46 89	40 10	0 85	14 75	17 25
corrected	4	55	50	1832	458 00	36 63	33 30	0 90	12 50	13.75
MIDDLE ATLANTIC										
vector	8			5019	827 37					
converted	8	135	109	5001	825 12	45 88	37 04	0.80	13.62	16.87
corrected	8	122	105	4063	507.87	38 69	33 30	0 86	13 12	15.25
NORTH WEST										
vector	3			2368	789 33					
converted	3	120	115	2381	787.00	20 53	19 67	0 95	38 33	40 00
corrected	3	119	114	1917	639 00	16 81	16 24	0 96	38 00	39 33
SOUTH										
vector	8			6390	798.75					
converted	8	161	140	8388	795.75	45 47	39 54	0 86	17 50	21 12
corrected	8	154	140	4955	619 37	35 39	32 17	0 90	17.50	19.25
MID WEST										
vector	15			12228	764.25					
converted	16	425	312	12189	761.81	39.06	28 68	0 73	19.50	26.56
corrected	16	326	312	7551	471 93	24 20	23 16	0 95	19 50	20 37
NEWENGLAND										
vector	6			1830	305 00					
converted	6	52	42	1816	302 66	43.23	34 92	0.80	7.00	8.66
corrected	6	49	40	1486	247 66	37.15	30 32	0 81	6 66	8.16
WASHDC										
vector	337			8825	24.49					
converted	337	2127	1295	7486	22 21	5 78	3 51	0.60	3 84	6.31
corrected	337	1533	1037	7247	21.50	6 98	4.72	0.67	3 07	4.54

Figure 4-2: Database Complexity For WASHDC And USA Databases

We divided the USA database into six zones and compiled each into a separate arc-node representation. This was primarily to look for variations within the database. As a group there was rather good consistency when compared to the statistics for the WASHDC database. For each area in Figure 4-2, the USA zones and WASHDC, statistics were computered at three points. The first point shown in the row labeled *vector* was complexity of the original vector data. The second (*converted*) was computed after the conversion to arc-node format, and the third (*corrected*) was after automatic detection and interactive correction of topological problems such as slivers, gaps, or closure problems. For the USA database one can observe that the number of points, nodes, number of segments, number of points per segment, and number of points per node decreased in each of the six zones. In addition, the percentage of points decreased more than the percentage of segments, and the percentage decrease in the number of segments was larger than decrease in the number of nodes. Figure 4-3 shows the number of points decreased more in the six regional zone data than in Washington D.C data. This is due to the large number of shared edges in the USA database. The number of segments and nodes decreased more in the Washington D.C. data set than in the regional data set since there were more occurances of slivers and gaps along shared boundaries which caused a large number of segments to be collapsed into a single segment.

```
                   seg :  nodes: points:
south west:        20.28  15.25  33.98
middle atlantic:    9.62   3.66  19.04
north west:         1.66   0.86  19.04
south:              4.34   0.00  22.45
mid west:          23.29   0.00  38.24
newengland:         5.76   4.76  18.79
------------------------------------
usa:               10.83   4.09  25.26
------------------------------------
washdc:            27.92  19.92  12.21
```

Figure 4-3: Percentage Reduction From Vector To Arc-Node Format

5. Conclusions

We have presented a brief description of the integration of multiple data representations within the MAPS system developed at Carnegie Mellon University. MAPS integrates schema-based representations of spatial knowledge, and multiple representations of spatial location using vector, arc-node, and hierarchical containment descriptions. We believe that the use of heterogenous representations tailored to particular data requirements or that capitalize on search or query efficiencies will be necessary if we are to reach our goal of intelligent spatial databases. Certainly this work is in sharp contrast with more homogeneous approaches such as regular decomposition (quadtree) and relational databases. There needs to be more testing and evaluation of prototype representation systems on realistic test databases as we attempt to design future spatial database systems.

6. References

1. McKeown, D.M., "MAPS: The Organization of a Spatial Database System Using Imagery, Terrain, and Map Data," *Proceedings: DARPA Image Understanding Workshop,* June 1983, pp. 105-127, Also available as Technical Report CMU-CS-83-136
2. McKeown, D.M.,, "Digital Cartography and Photo Interpretation from a Database Viewpoint," in *New Applications of Databases,* Gargarin, G. and Golembe, E., ed., Academic Press, New York, N. Y., 1984, pp. 19-42.
3. McKeown, D. M., "Knowledge-Based Aerial Photo Interpretation," *Photogrammetria, Journal of the International Society for Photogrammetry and Remote Sensing,* Vol. 39, 1984, pp. 91-123, Special Issue on Pattern Recognition
4. McKeown, D.M., "The Role of Artificial Intelligence in the Integration of Remotely Sensed Data with Geographic Information Systems," *IEEE Transactions on Geoscience and Remote Sensing,* Vol. GE-25, No. 4, July 1987, pp. to appear, Also available as Technical Report CMU-CS-86-174
5. Avron Barr and Edward A. Feigenbaum, *The Handbook of Artificial Intelligence,* William Kaufmann, Inc., Los Altos, CA, , Vol. 1, 1981.

0-937294-88-8