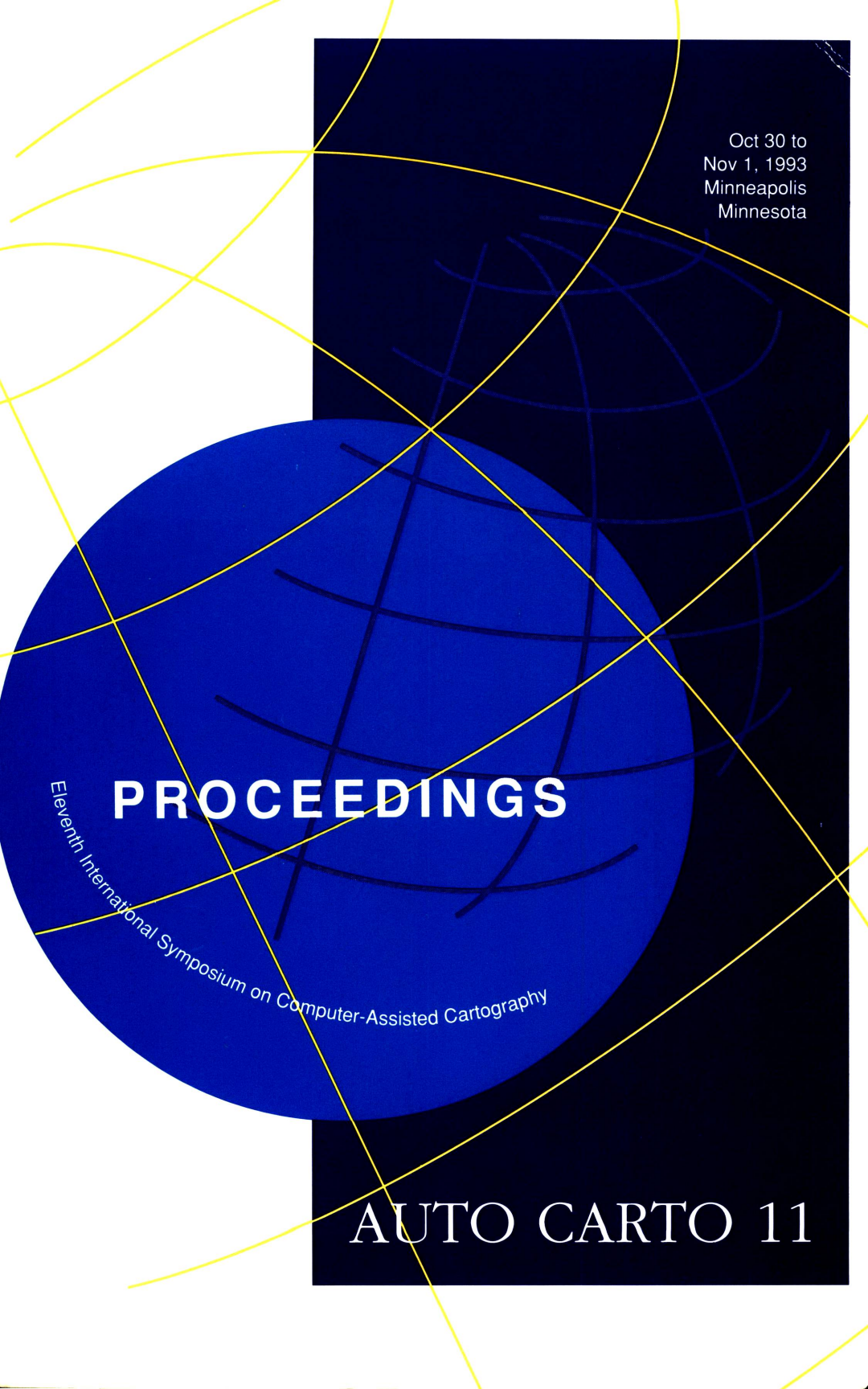

Oct 30 to
Nov 1, 1993
Minneapolis
Minnesota
PROCEEDINGS
Eleventh International Symposium on Computer-Assisted Cartography
AUTO CARTO 11

AUTO-CARTO 11 PROCEEDINGS

30 October - 1 November 1993

Minneapolis Convention Center

Minneapolis, Minnesota

TECHNICAL PAPERS
Eleventh International Symposium on
Computer-Assisted Cartography

ISBN 1-57083-001-0

Published by

American Society for Photogrammetry and Remote Sensing
American Congress on Surveying and Mapping
5410 Grosvenor Lane
Bethesda, Maryland 20814-2160

FOREWORD AND ACKNOWLEDGEMENTS

The Eleventh International Symposium on Computer-Assisted Cartography (Auto-Carto-11) continues the tradition of Auto Carto conferences dating back to the early 1970s. Auto-Carto has become internationally recognized for presenting state-of-the-art papers in digital cartography, the manipulation of spatial data, and geographic information systems. Traditionally, the Auto-Carto conferences have been held every other year and now are held in alternate years from the Spatial Data Handling conferences, sponsored by the IGU. Both of these conferences--Auto-Carto and Spatial Data Handling--focus on basic conceptual/theoretical research. This conference, scheduled at the Minneapolis Hilton from October 30th to November 1st, 1993, is being held directly before GIS/LIS'93. The forty-three papers accepted for publication in this volume were selected from over seventy abstracts submitted to the program committee in April of 1993.

We would like to thank the many individuals who have helped with the organization and planning for Auto-Carto-11. Since the original planning for the conference only began during GIS/LIS'92, it was necessary to put together a program in a short period of time. We wish to thank John Lisack, Executive Director of ACSM and Bill French, Executive Director of ASPRS, as well as the Joint Convention Advisory Committee, for approving and supporting the concept of an Auto-Carto-11. Linda Hachero and Cheryl Hill of Fait Accompli efficiently took care of local arrangements in Minneapolis, as well as distribution of program materials. We also wish to thank Jodi Larson at The University of Minnesota for her careful attention in sending out the conference correspondence. The Program Committee of Barbara Buttenfield, Nick Chrisman, Keith Clarke, Max Egenhofer, Gail Langran, David Mark, Mark Monmonier, Jean-Claude Muller, and Rob Weibel carefully reviewed the abstracts and provided valuable feedback on the final program.

This volume contains papers, presented at the conference, on an array of topics related to digital cartography and GIS, including spatial theory, hypermedia, generalization, visualization, and algorithmic development. We hope you will find it a valuable addition to your professional library.

Robert B. McMaster
The University of Minnesota

Marc P. Armstrong
The University of Iowa

TABLE OF CONTENTS

Spatial Theory II

Multiple Representations

Visualization I

Terrain Representation

Algorithmic Issues I

Three-Dimensional Modeling

Multimedia/Hypermedia/Graphics

Generalization

Parallel Computing

AUTHOR INDEX

A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis*

Max J. Egenhofer and Jayant Sharma
National Center for Geographic Information and Analysis
and
Department of Surveying Engineering
Department of Computer Science
University of Maine
Boardman Hall
Orono, ME 04469-5711, U.S.A.
{max, jayant}@grouse.umesve.maine.edu

and

David M. Mark
National Center for Geographic Information and Analysis
and
Department of Geography
State University of New York at Buffalo
Buffalo, NY 14261-0023, U.S.A.
geodmm@ubvms.cc.buffalo.edu

Abstract

Two formalisms for binary topological spatial relations are compared for their expressive power. The *4-intersection* considers the two objects' interiors and boundaries and analyzes the intersections of these four object parts for their content (i.e., emptiness and non-emptiness). The *9-intersection* adds to the 4-intersection the intersections with the two objects' complements. The major results are (1) for objects with co-dimension 0, the 4-intersection and the 9-intersection with the content invariant provide the same results; and (2) for objects with co-dimension > 0, the 9-intersection with the content invariant provides more details than the 4-intersection. These additional details are crucial to determine when two objects are equal. It is also demonstrated that the additional details can provide crucial information when specifying the semantics of spatial relations in GIS query languages.

Introduction

During the last three years, the formal description of spatial relations has received unprecedented attention in the GIS arena. The focus of many investigations was on a particular formalism to represent *topological relations* (Egenhofer and Franzosa 1991; Herring 1991; Pigot 1991; Hadzilacos and Tryfona 1992; Hazelton *et al.* 1992; Clementini *et al.* 1993; Cui *et al.* 1993; Wazinski 1993). Complementary activities in the area of cardinal directions (Peuquet and Ci-Xiang 1987; Frank 1992; Freksa 1992; Papadias and

* This work was partially supported through the NCGIA by NSF grant No. SES-8810917. Additionally, Max Egenhofer's work is also supported by NSF grant No. IRI-9309230, a grant from Intergraph Corporation, and a University of Maine Summer Faculty Research Grant. Some of the ideas were refined while on a leave of absence at the Università di L'Aquila, Italy, partially supported by the Italian National Council of Research (CNR) under grant No. 92.01574.PF69. Jayant Sharma is partially supported by a University of Maine Graduate Research Assistantship (UGRA).

Sellis 1992; Jungert 1992) exist, however, unlike the studies of topological relations, formalizations of cardinal directions are based on a diversity of models. This paper focuses on the two primary models used for binary topological relations, the 4-intersection and the 9-intersection, which is an extension of the 4-intersection.

The initial model for binary topological relations, developed for two 2-dimensional objects embedded in $\mathbb{R}^2$, compared the boundaries and interiors of the two objects and classified the relations by whether the intersections of these four parts were empty or not (Egenhofer 1989; Egenhofer and Herring 1990; Egenhofer and Franzosa 1991). This model is called the *4-intersection.* An extension of the 4-intersection includes also the intersections with the exteriors, and allows for the identification of more detailed relations, particularly if one or both objects are embedded in higher-dimensional spaces, such as the topological relation between a line and an areal object or the topological relation between two lines in $\mathbb{R}^2$ (Egenhofer and Herring 1991). This model is called the *9-intersection.*

The need for the more extensive 9-intersection has been questioned by several researchers who have tried to model line-region and line-line relations in $\mathbb{R}^2$ just with the 4-intersection (Svensson and Zhexue 1991; Hadzilacos and Tryfona 1992; Hazelton *et al.* 1992; Clementini *et al.* 1993). This paper demonstrates that the 4-intersection and 9-intersection reveal the same results only if both objects are n-dimensional and embedded in $\mathbb{R}^n$ such that different between the dimensions of the embedding space and the objects is 0. These objects are said to have co-dimension 0. For all other configurations with co-dimension > 0, such as the relations between a line and a region in $\mathbb{R}^2$ or the relations between two lines in $\mathbb{R}^2$, it is shown that the 9-intersection distinguishes among topological relations that would be considered the same using the 4-intersection model.

The remainder of this paper is structured as follows: The next section briefly reviews the 4-intersection and the 9-intersection models for topological relations. Then the consequences of using the 4-intersection or 9-intersection are elaborated for line-region and line-line relations in $\mathbb{R}^2$. A discussion of using alternatives to the boundary/interior 4-intersections completes the comparison of different models for binary topological relations. The conclusions provide a concise summary of the results and their importance.

Models for Topological Relations

4-Intersection

Binary topological relations between two objects, A and B, are defined in terms of the four intersections of A's boundary (∂A) and interior (A°) with the boundary (∂B) and interior (B°) of B (Egenhofer and Franzosa 1991). This model is concisely represented by a 2×2-matrix, called the *4-intersection.*

$$\mathfrak{I}_4(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B \\ \partial A \cap B^\circ & \partial A \cap \partial B \end{pmatrix} \quad (1)$$

Topological invariants of these four intersections, i.e., properties that are preserved under topological transformations, are used to categorize topological relations. Examples of topological invariants, applicable to the 4-intersection, are the content (i.e., emptiness or non-emptiness) of a set, the dimension, and the number of separations (Franzosa and Egenhofer 1992). The content invariant is the most general criterion as other invariants can be considered refinements of non-empty intersections. By considering the values empty ($\varnothing$) and non-empty ($\neg\varnothing$) for the four intersections, one can distinguish $2^4 = 16$ binary topological relations. Eight of these sixteen relations can be realized for homogeneously

2-dimensional objects with connected boundaries, called *regions*, if the objects are embedded in $\mathbb{R}^2$ (Egenhofer and Herring 1990) (Figure 1).

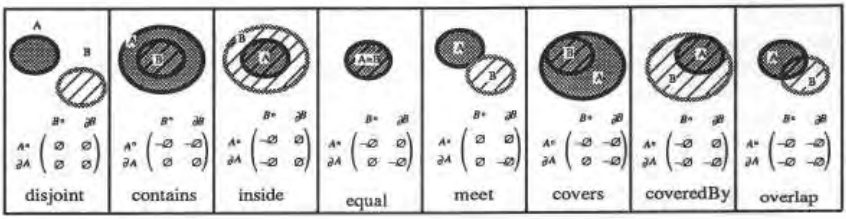


Figure 1: Examples of the eight topological relations between two regions in $\mathbb{R}^2$.

Also eight topological relations can be found between two *lines* in $\mathbb{R}^1$ (Pullar and Egenhofer 1988) (Figure 2).The latter set of relations corresponds to Allen's interval relations (Allen 1983) if the order of $\mathbb{R}^1$ is disregarded. With the exception of *overlap*, the two sets of 4-intersections for region-region relations in $\mathbb{R}^2$ and line-line relations in $\mathbb{R}^1$ are identical. The difference is due to the fact that regions have *connected* boundaries, while lines have *disconnected* boundaries; therefore, for a region whose boundary intersects with the other region's interior *and* exterior, its boundary must also intersect with the other region's boundary. This conclusion cannot be drawn for two lines because their boundaries are disconnected.

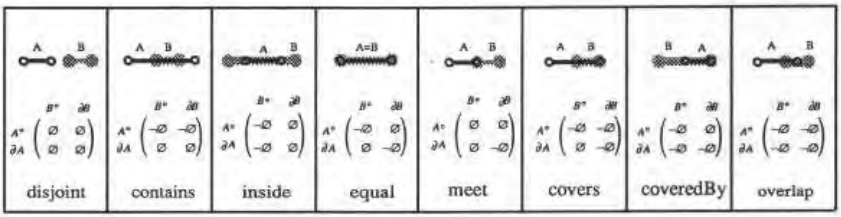


Figure 2: Examples of the eight topological relations between two lines in $\mathbb{R}^1$.

9-Intersection

The 4-intersection model is extended by considering the location of each interior and boundary with respect to the other object's exterior; therefore, the binary topological relation between two objects, A and B, in $\mathbb{R}^2$ is based upon the intersection of A's interior (A°), boundary (∂A), and exterior (A^-) with B's interior (B°), boundary (∂B), and exterior (B^-). The nine intersections between the six object parts describe a topological relation and can be concisely represented by a 3×3-matrix $\Im_9$, called the *9-intersection*.

$$\Im_9(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix} \qquad (2)$$

In analogy to the 4-intersection, each intersection will be characterized by a value empty ($\varnothing$) or non-empty ($\neg\varnothing$), which allows one to distinguish $2^9 = 512$ different configurations. Only a small subset of them can be realized between two object in $\mathbb{R}^2$.

The relations that can be realized depend on particular topological properties of the objects involved and their relationship to the embedding space. For example, the boundary of a spatial region in $\mathbb{R}^2$ is a Jordan Curve (separating the interior from the exterior). On the other hand, the boundary of a simple line consists of two nodes, and unlike a region's boundary in $\mathbb{R}^2$, a line's boundary in $\mathbb{R}^2$ does not separate the interior from the exterior. These topological properties of the objects have to be considered when investigating which empty/non-empty 9-intersection can be realized. For this goal, we formalized for each combination of regions and lines embedded in $\mathbb{R}^2$ a set of properties as conditions for binary topological relations, that must hold between the parts of the two objects (Egenhofer and Herring 1991). These properties can be expressed as consistency constraints in terms of the 9-intersection, such that by successively eliminating from the set of 512 relations those relations that violate a consistency constraint, one retains the candidates for those 9-intersections that can be realized for the particular spatial data model (Egenhofer and Sharma 1993). The existence of these relations is then proven by finding geometric interpretations for the corresponding 9-intersections.

Existing 9-Intersections Between Two Regions in $\mathbb{R}^2$: With the 9-intersection, the same set of region-region relations can be found as for the 4-intersection (Egenhofer and Franzosa 1991). No additional relations due to the consideration of exterior-intersections are possible.

Existing 9-Intersections Between Two Lines in $\mathbb{R}^2$: As expected, the 9-intersection reveals the same number of line-line relations in $\mathbb{R}^1$ as the 4-intersection; however, in $\mathbb{R}^2$, the 9-intersection identifies another 25 relations for relations between two simple lines (i.e., lines with exactly two end points) (Egenhofer 1993). Another 21 relations are found if the lines can be branched so that they have more than two end points (Egenhofer and Herring 1991).

Existing 9-Intersections Between A Line and a Region in $\mathbb{R}^2$: With the 9-intersection, 19 topological relations between a simple line and a region in $\mathbb{R}^2$ can be found (Mark and Egenhofer 1992), and a 20th configuration if the line is branched (Egenhofer and Herring 1991).

Need for 9-Intersection

Since the realization of existing topological relations in both models is based on particular topological properties of the objects and the relationship to the embedding space, the following generalizations can be made:

- If the two objects are simply connected, their boundaries form Jordan Curves (or the corresponding configurations in higher-dimensional spaces), and the objects have co-dimension 0, then the same eight topological relations can be realized as between two regions in $\mathbb{R}^2$. For example, the same relations as between two regions in $\mathbb{R}^2$ also exist between two volumes.

- If the two objects are simply connected, each boundary forms a separation, and the objects have co-dimension 0, then the same eight topological relations can be realized as between two lines in $\mathbb{R}^1$.

If the co-dimension constraint is relaxed such that one or both objects can be embedded in a higher-dimensional space, due to the greater degree of freedom, the objects may take additional configurations that are not represented by one of the relations between objects with co-dimension 0. For example, if two lines "cross," they have non-empty interior-interior intersections, while the other three intersections are empty. Such a 4-intersection

cannot be realized for two lines in $\mathbb{R}^1$. On the other hand, some 4-intersections may have ambiguous geometric interpretations. From a practical point of view, there are certainly situations in which one would like to distinguish between them when querying a geographic database. Therefore, in general, a different model is necessary to account also for relations involving n-dimensional objects that are embedded in $\mathbb{R}^m$, $m > n$. Our focus is on $m = 2$ and its topological relations with objects of dimension $n = 1$—topological relations with points ($n = 0$) are trivial.

To analyze the differences between the two methods, the conceptual neighborhoods of each set of relations will be used. *Conceptual neighborhoods* organize a set of relations in a diagram such that similar relations are close to each other. The computational tool to identify conceptual neighborhoods is the topology distance (Egenhofer and Al-Taha 1992) which calculates the differences between two empty/non-empty 9-intersections. Pairs of relations with the least number of non-zero differences are considered to be conceptual neighbors. Conceptual neighborhoods are represented as a graph in which the relations are nodes and the conceptual neighbors are edges between the nodes.

Line-Region Relations

If one removes for each of these nineteen 9-intersections the entries of the exterior intersections, what remains is a 4-intersection based on boundary/interior intersections. A straightforward inspection reveals that only six of the 19 line-region relations are uniquely characterized by the 4-intersection. These relations are all located along the edge of the conceptual neighborhood diagram shown in Figure 3 (A1, B1, C1, C5, D5, and E5). The remaining thirteen cases can be grouped into five distinct groups, each having a characteristic 4-intersection: A2, A3, and A5, called the A-band; and the B-, C-, D-, and E-band with B2-B3, C2-C4, D2-D4, and E3-E4, respectively. The distinguishing factor among the configurations in each group is whether the interior or boundary of the line intersects the exterior of the region. Since the intersections with the exterior are not considered, the 4-intersection is the same for all configurations in any band.

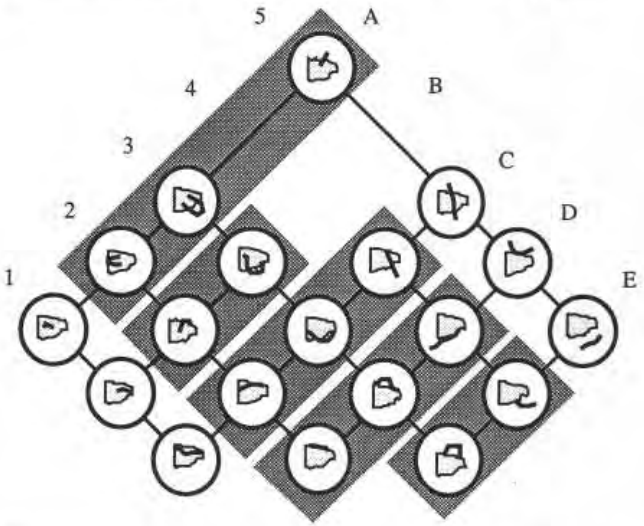


Figure 3: The conceptual neighborhoods of topological line-region relations in $\mathbb{R}^2$. Groups of relations with the same 4-intersection are shaded.

In the data model the interior, boundary, and exterior are distinct topological entities. Hence each configuration, in any group in Figure 4, is topologically distinct from the other configurations in the same group. That is, there is no topological transformation that converts one configuration to another. The 4-intersection, however, cannot distinguish between them and thus it is insufficient.

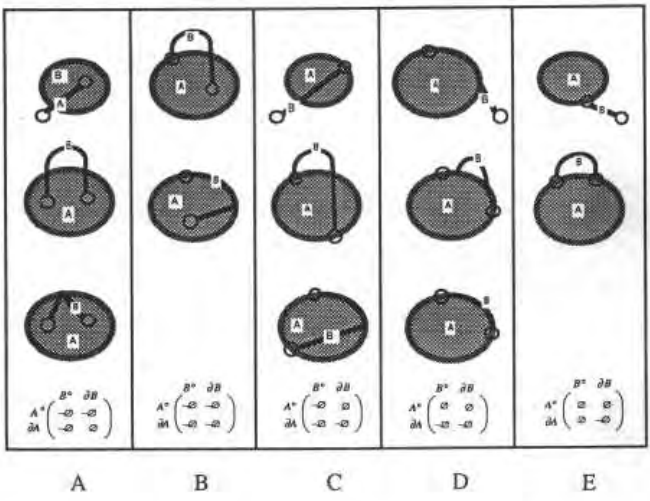


Figure 4: The five groups of line-region relations. Relations in each group have the same 4-intersection, but distinct 9-intersections.

The importance of the additional information available in the 9-intersection becomes more obvious when one investigates the meaning of certain spatial predicates. Spatial predicates are commonly used as selection criteria in GIS queries and in order to process such queries, the semantics of the terms have to be formalized. If one considers the line and the region to be a road and a park, respectively, then one may consider the meaning of the spatial predicate "enters" as "the line has to have parts inside and outside of the region." Based on the 9-intersection, the six configurations with non-empty interior-interior, interior-boundary, and interior-exterior intersections qualify for this constraint—A3, A5, B3, and C3, C4, C5. Therefore, such a definition of "enters" splits the shaded bands (A, B, and C), i.e., the configurations that cannot be distinguished by the 4-intersection. Based solely on the 4-intersection, this distinction would not have been possible, because the set of relations with non-empty interior-interior and interior-boundary intersections includes three configurations with non-empty interior-exterior intersections; therefore, using the 4-intersection or the 9-intersection as the underlying model to process such a query, one may get considerably different results, some of which would contradict the definition of the term "enters."

The question remains open whether the 4-intersection would be sufficient if one had particular knowledge about the objects' geometric properties such as whether the lines are straight or possibly curved, and whether the regions are convex or possibly concave. First, the knowledge of only one such geometric property does not influence the existence of topological relations. For example, if one fixes the shape of the line to be straight then the region can be deformed, where necessary, to a concave object so that all 19 relations can be realized. Likewise, if the region were fixed to be convex, one could bend the line so that all 19 relations can be realized. The case is, however, different if both objects are constrained.

Among the 19 line-region relations, only 11 can be found for a straight line and a convex region. The eight additional ones for curved lines or convex regions all fall in the range of relations that cannot be distinguished with the 4-intersection: they are A2 and A3, B2 and B3, C2 and C3, D3, and E3. With the exception of the relations in band D, there is a 1:1 mapping between the 4-intersection- and 9-intersection-relations for a straight line and a convex region (the relations in band B cannot be realized in either model). The two straight-line-convex-region-relations that cannot be distinguished are (1) a straight line is completely in the boundary of a convex region and (2) a straight line starts in the boundary following the boundary for a while, until it ends in the convex region's exterior.

Line-Line Relations

The 9-intersection characterization results in 33 distinct line-line relations in $\mathbb{R}^2$ (Egenhofer and Herring 1991). Since the 4-intersection can only characterize $2^4 = 16$ distinct relations, it is obvious that the 9-intersection provides a much finer resolution. Figure 5 shows a subset of the conceptual neighborhood of topological line-line relations in $\mathbb{R}^2$ and highlights the groups of those relations that would not be distinguished by the 4-intersection. There are 10 such groups containing between two and four relations. Six relations from the 4-intersection have exactly one corresponding 9-intersection relation.

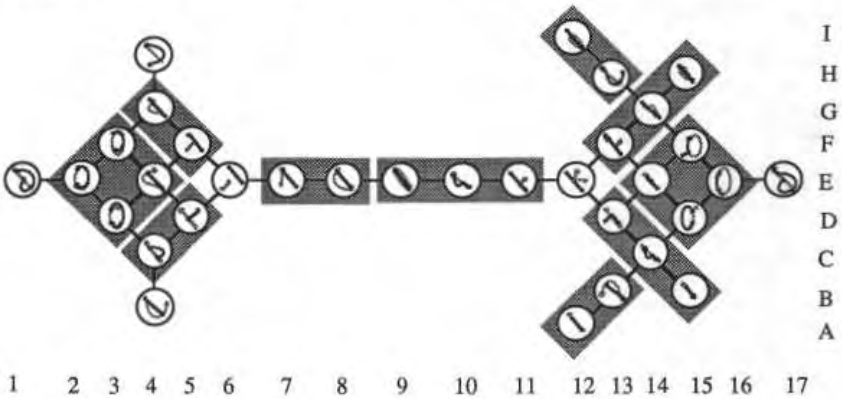
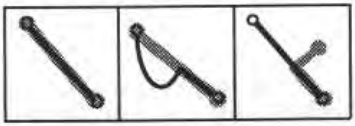


Figure 5: A subset of the conceptual neighborhood of topological line-line relations in $\mathbb{R}^2$ (not all links of topology distance 1 are depicted). Groups of relations with the same 4-intersection are shaded.

As a strongly motivating example for the need of the finer granularity of the 9-intersection, consider the topological relation *equal* (E9). Equal is part of a group with another two relations, all of which have the same 4-intersection (Figure 6), and not a singleton as one would expect. Using the 9-intersection, only the configuration in Figure 6a would be classified as an example of *equal*.



(a) (b) (c)

Figure 6: Examples of topological relations between two lines in $\mathbb{R}^2$ that have the same 4-intersection ($A^\circ \cap B^\circ = \neg\emptyset$; $A^\circ \cap \partial B = \emptyset$; $\partial A \cap B^\circ = \emptyset$; and $\partial A \cap \partial B = \neg\emptyset$), but different 9-intersections as their boundary-exterior and interior-exterior intersections differ.

Alternative 4-Intersections

The data model used here is based on concepts from point-set topology. A spatial region is a simply connected area whose boundary is a Jordan curve; therefore, it has three topologically distinct parts: the interior, boundary, and exterior. Since a region is a 2-dimensional object in a 2-dimensional space, specifying any one part completely determines the region and its other parts.

Based on this observation it appears reasonable to assume that topological relations between regions can be characterized by considering the intersections of any pair of parts, for example, boundary/exterior or interior/exterior, rather than only the boundary/interior intersections. To assess such alternatives, one has to determine whether the 4-intersection based on the boundary/interior intersections is equivalent to one based on boundary/exterior or interior/exterior intersections. If so, the characterization of topological relations would have to be the same in each case.

- A 4-intersection based on boundary/exterior intersections cannot express the distinction between the relations *meet* and *overlap*. The reason is that the only difference between meet and overlap is whether the interiors do not or do intersect, respectively. Since the intersections of interiors is not considered, the 4-intersections, for the configurations called meet and overlap in Figure 1, are exactly the same.

- Similarly, a 4-intersection based on interior/exterior intersections cannot express the distinction between the pairs of relations: *meet* and *disjoint*, *contains* and *covers*, *inside* and *coveredBy*, because the only difference in each case is whether the boundaries intersect or not. Since the intersection of boundaries is not considered, the 4-intersections are exactly the same.

- Finally, the alternatives of using a 4-intersection based on the closure—the union of the interior and boundary—in combination with the interior, boundary, or exterior reveal the same deficiencies as they cannot distinguish between overlap and covers/coveredBy, or overlap and meet.

The conclusion is therefore that only boundary and interior should be used for the 4-intersection in characterizing topological relationships between regions.

Conclusions

The two primary models of topological relations, the 4-intersection and the 9-intersection, were compared for their expressive powers. Table 1 summarizes the results of the numbers of relations that can be realized in each model for co-dimension 0. It was shown in this

paper that for co-dimension 0 exactly the same relations can be realized with the 4- and 9-intersection.

co-dimension 0	**region**	**line**
region	$\Im_4$: 8 relations $\Im_9$: 8 relations	N/A
line	N/A	$\Im_4$: 8 relations $\Im_9$: 8 relations

Table 1: Number of binary topological relations that can be realized for regions and lines in co-dimension 0 with the 4-intersection and the 9-intersection.

The situation is quite different if the two objects are embedded in a higher-dimensional space (Table 2). The 9-intersection has a finer granularity to distinguish relations between a line and a region, and between two lines embedded in $\mathbb{R}^2$. The most crucial difference was found for line-line relations, where the 4-intersection applied to $\mathbb{R}^2$ does not provide a useful definition of an "equal" relation. On the other hand, the 9-intersection compensates this shortcoming.

co-dimension 1	**line**
	straight line
region	$\Im_4$: 11 relations $\Im_9$: 19 relations
convex region	*$\Im_4$: 10 relations* *$\Im_9$: 11 relations*
line	$\Im_4$: 16 relations $\Im_9$: 33 relations
straight line	*$\Im_4$: 11 relations* *$\Im_9$: 11 relations*

Table 2: Number of binary topological relations that can be realized for regions and lines in co-dimension 1 with the 4-intersection and the 9-intersection.

The results have an impact on the implementation of spatial relations in a GIS. Although the 9-intersection is necessary to distinguish such details, not all nine intersections have to be calculated at all times when processing a query with such a topological relation. Most obvious is this in the case of the line-region relations, where all intersections between the line's exterior and the three parts of the region are non-empty, independent of the relation between the two objects; therefore, calculating these three intersections would not provide any information about the particular configuration.

The results of this paper must be considered in combination with results obtained from human-subject testing of topological relations (Mark and Egenhofer 1992). Initial studies of line-region configurations showed there that the differences in the distinctions made by the 9-intersection are sometimes crucial when humans select natural-language terminology to describe some spatial situations. Only if the present analysis is considered in the entirety of the *interplay* between formal mathematics and human-subjects testing, its significance will become obvious.

Acknowledgments

Over the years, a number of colleagues and friends have contributed to and participated in this research. Particularly, discussions with John Herring, Bob Franzosa, Andrew Frank, Christian Freksa, Tony Cohn, Eliseo Clementini, and Paolino di Felice helped us in getting

a better understanding of the nature of spatial relations. Thanks also to Kathleen Hornsby who helped with the preparation of the manuscript.

References

J. F. Allen (1983) Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11): 832-843.

E. Clementini, P. Di Felice, and P. van Oosterom (1993) A Small Set of Formal Topological Relationships Suitable for End-User Interaction. in: D. Abel and B. C. Ooi (Eds.), *Third International Symposium on Large Spatial Databases, SSD '93. Lecture Notes in Computer Science* 692, pp. 277-295, Springer-Verlag, New York, NY.

Z. Cui, A. Cohn, and D. Randell (1993) Qualitative and Topological Relationships in Spatial Databases. in: D. Abel and B. Ooi (Eds.), *Third International Symposium on Large Spatial Databases. Lecture Notes in Computer Science* 692, pp. 296-315, Springer-Verlag, New York, NY.

M. Egenhofer (1989) A Formal Definition of Binary Topological Relationships. in: W. Litwin and H.-J. Schek (Ed.), *Third International Conference on Foundations of Data Organization and Algorithms (FODO). Lecture Notes in Computer Science* 367, pp. 457-472, Springer-Verlag, New York, NY.

M. Egenhofer (1993) Definitions of Line-Line Relations for Geographic Databases, *IEEE Data Engineering* 16 (in press).

M. Egenhofer and K. Al-Taha (1992) Reasoning About Gradual Changes of Topological Relationships. in: A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Models of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science* 639, pp. 196-219, Springer-Verlag, New York, NY.

M. Egenhofer and R. Franzosa (1991) Point-Set Topological Spatial Relations. *International Journal of Geographical Information Systems* 5(2): 161-174.

M. Egenhofer and J. Herring (1990) A Mathematical Framework for the Definition of Topological Relationships. *Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp. 803-813.

M. Egenhofer and J. Herring (1991) *Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases*. Technical Report, Department of Surveying Engineering, University of Maine, Orono, ME.

M. Egenhofer and J. Sharma (1993) Topological Relations between Regions in R^2 and Z^2. in: D. Abel and B. Ooi (Eds.), *Third International Symposium on Large Spatial Databases. Lecture Notes in Computer Science* 692, pp. 316-336, Springer-Verlag, New York, NY.

A. Frank (1992) Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing* 3(4): 343-371.

R. Franzosa and M. Egenhofer (1992) Topological Spatial Relations Based on Components and Dimensions of Set Intersections. *SPIE's OE/Technology '92—Vision Geometry*, Boston, MA, pp. 236-246.

C. Freksa (1992) Using Orientation Information for Qualitative Spatial Reasoning. in: A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Models of Spatio-Temporal*

Reasoning in Geographic Space. Lecture Notes in Computer Science 639, pp. 162-178, Springer-Verlag, New York, NY.

T. Hadzilacos and N. Tryfona (1992) A Model for Expressing Topological Integrity Constraints in Geographic Databases. in: A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Models of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science* 639, pp. 252-268, Springer-Verlag, Pisa.

N. W. Hazelton, L. Bennett, and J. Masel (1992) Topological Structures for 4-Dimensional Geographic Information Systems. *Computers, Environment, and Urban Systems* 16(3): 227-237.

J. Herring (1991) The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems. in: D. Mark and A. Frank (Ed.), *Cognitive and Linguistic Aspects of Geographic Space.* pp. 313-350, Kluwer Academic Publishers, Dordrecht.

E. Jungert (1992) The Observer's Point of View: An Extension of Symbolic Projections. in: A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Models of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science* 639, pp. 179-195, Springer-Verlag, New York, NY.

D. Mark and M. Egenhofer (1992) An Evaluation of the 9-Intersection for Region-Line Relations. *GIS/LIS '92*, San Jose, CA, pp. 513-521.

D. Papadias and T. Sellis (1992) Spatial Reasoning Using Symbolic Arrays. in: A. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Models of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science* 639, pp. 153-161, Springer-Verlag, New York, NY.

D. J. Peuquet and Z. Ci-Xiang (1987) An Algorithm to Determine the Directional Relationship Between Arbitrarily-Shaped Polygons in the Plane. *Pattern Recognition* 20(1): 65-74.

S. Pigot (1991) Topological Models for 3D Spatial Information Systems. *Autocarto 10*, Baltimore, MD, pp. 368-392.

D. Pullar and M. Egenhofer (1988) Towards Formal Definitions of Topological Relations Among Spatial Objects. in: D. Marble (Ed.), *Third International Symposium on Spatial Data Handling*, Sydney, Australia, pp. 225-242.

P. Svensson and H. Zhexue (1991) Geo-SAL: A Query Language for Spatial Data Analysis. in: O. Günther and H.-J. Schek (Eds.), *Advances in Spatial Databases—Second Symposium, SSD '91. Lecture Notes in Computer Science* 525, pp. 119-140, Springer-Verlag, New York, NY.

P. Wazinski (1993) Graduated Topological Relations. Technical Report 54, University of the Saarland, Saarbrücken, Germany.

SUPPORTING VISUAL INTERACTIVE LOCATIONAL ANALYSIS USING MULTIPLE ABSTRACTED TOPOLOGICAL STRUCTURES

Paul J. Densham
National Center for Geographic Information and Analysis
Department of Geography
State University of New York at Buffalo
Buffalo, New York 14261-0023
densham@geog.buffalo.edu

and

Marc P. Armstrong
Departments of Geography and Computer Science
The University of Iowa
316 Jessup Hall
Iowa City, Iowa 52242
armstrng@umaxc.weeg.uiowa.edu

ABSTRACT

Data structures for automated cartography traditionally have been based on either vector or tessellation models. We describe a set of topological abstractions, derived from a vector approach to the representation of geographic space, that were developed to support the interactive solution of location-selection problems. When augmented with an appropriate representation of geometry, these abstractions are used to generate cartographic displays that support interactive decision-making. The advantages of this approach include: the use of the same data abstractions for analysis and display purposes; support for multiple representations of networks and, therefore, a degree of scale independence; and, finally, support for highly interactive problem-solving and decision-making because map generation can be decomposed into parallel processes.

1.0 INTRODUCTION

Decision-makers faced with ill-structured social and environmental spatial problems increasingly are adopting spatial decision support systems (SDSS) to help them find solutions. When decision-makers try to resolve such problems, they can work within four spaces (Densham and Goodchild, 1990).

- *Objective space* contains all feasible options available for a given problem.
- *Decision space* is bounded by the decision-maker's knowledge, objectives and values. It contains those options considered feasible by the decision-maker.
- *Model space* is determined, for a given model, by its variables and their values, its parameter values, and the relationships represented in the model.
- *Geographic space* provides a context for a decision and is depicted using cartographic and other graphical representations. Because of its richness and complexity, geographical space is abstracted and represented in objective, decision and model spaces.

For any given problem, SDSS users normally wish to use the system to help them explore their decision space. Because an individual's decision space typically is some subset of objective space, the challenge facing SDSS designers is to empower their

users and to expand their decision space to include much of the objective space. One approach to this challenge is to provide users with modelling capabilities and complementary representations of geographic space; in concert, these representations enable users to evaluate solutions from models using their expertise, objectives and values by synthesizing mathematical and cartographic representations of a problem.

Despite the emphasis on exploring decision spaces in the SDSS literature, many current systems support a process of decision-making in which graphical representations play a very restricted role (Densham, 1993a). This process often follows more traditional modes of GIS use, in which a system's capabilities are sequenced in a linear, "work-flow" approach that results in a final graphical product. To explore a decision space, however, decision-makers require a series of capabilities, consisting of both analytical and graphical representations, that can be combined in flexible sequences. Moreover, decision-makers increasingly wish to interact with a system through any of the representations of a problem that it provides. For example, in our applied work with decision-makers we often display a map of a locational configuration and listen as they articulate questions. In most cases, such questions would be answered if the user could manipulate the graphic directly - by dragging a facility to a new location, by adding or removing a facility, or making some other change - and the system would respond by invoking the appropriate analytical capability. While such linkages are very similar to those found in the direct manipulation interfaces of word processors and object-based drawing packages, they typically are missing from geoprocessing software. Consequently, we listen as decision-makers articulate their questions and move their fingers across the screen to illustrate the kinds of actions they would like to make; we then try to translate these questions into an appropriate sequence of analytical operations.

In this paper, we show how multiple topological structures can be used to represent the analytical and cartographic components of a range of location-selection problems. These structures support bi-directional linkages between the analytical and cartographic representations of problems. As such, they provide a foundation for building a SDSS that supports a visual and interactive approach to decision-making; one that synthesizes the goal-seeking nature of traditional analytical modelling with the intuitive "what-if" approach supported by GIS.

2.0 REPRESENTATIONS

Historically, there have been conceptual and practical issues that have conditioned approaches to the representation of geographical relationships. These issues have been addressed in different ways by cartographers and spatial analysts. One dimension on which to measure such differences is the level of abstraction used in representing space, spatial relationships and thematic information. Cartographers have attempted to provide map users with appropriate, and often detailed, geometrical representations of space. Spatial analysts, on the other hand, have tended to employ highly abstracted representations of spatial relationships in their models. In extreme cases, only an abstracted form of topology is maintained and geometry is discarded completely - the origin-destination matrix is a prime example. While these abstracted relationships often are derived from detailed cartographic information, the extreme parsimony of the derived information has tended to preclude effective display. As a consequence of these differences, separate data structures have been developed for graphical and analytical purposes.

Cartographic data structures have benefitted from many years of development in the fields of automated cartography and GIS (e.g. Peucker and Chrisman, 1975). In contrast, analytical data structures reflect the "black-box" nature of modelling tools: analytical structures are often *ad hoc* and code-specific with little or no attention paid to placing them in a broader representational context. This results, in part, from the "stand-alone" nature of many custom-written modelling tools and the often scant attention paid to visualization by spatial analysts. Furthermore, the use of commercial modelling and analysis packages (including SAS, SPSS and GAUSS) often forces

spatial analysts to remove the geometry from their problems to make them fit the available representations and structures. Paradoxically, it is the very domain-independence of their representations and structures that makes these packages feasible commercially. In contrast, cartographic data structures have evolved to reflect and accommodate the standard set of spatial primitives around which most GIS are built. Thus, points, lines, areas, and raster cells are well-defined, and their representations in GIS that support both operations and display have been refined over twenty years. Despite these refinements, recent changes in software technology have caused researchers to view these structures as collections of objects. Unfortunately, spatial analysts have paid little attention to the identification and definition of the spatial primitives and objects which underlie their algorithms and models. Consequently, analytical objects normally are not the same as display objects in either their structure, content or function. This variation has, in part, led to an asymmetry: the display of cartographic objects must reflect the current status of analytical objects (e.g. which ones are facilities) but analytical objects typically carry little or no geometrical information and often only abstracted topological information that can be used to link them to cartographic objects.

3.0 VISUAL INTERACTIVE LOCATIONAL ANALYSIS

Decision-makers increasingly work in microcomputer environments for which a plethora of word-processors, spreadsheets, drawing and charting packages, and database management systems have been developed to support multiple forms of user interaction. Such software increasingly links graphics to other forms of representation and provides mechanisms for their direct manipulation, including drag-and-drop editing. For example, if a user changes the cost of a raw material in a spreadsheet cell, a graphic that depicts forecasted profits is updated automatically. The power of complementary analytical and graphical representations that have been designed to enhance a decision-maker's understanding of a problem has been recognized in many disciplines. Operations researchers, for example, have developed graphical methods for both formulating and interacting with analytical models. These visual interactive modelling (VIM, Hurrion, 1986) tools typically support bi-directional linkages among representations. Thus, using a spreadsheet analogy, users could either change the value of a cell and see the graphic updated, or they could manipulate the graphic and see the cells in the spreadsheet updated. In a GIS, maps are used to support spatial query. This form of linkage, however, normally is found only between the database and graphics capabilities of a GIS and bi-directional linkages typically are not a part of systems that couple GIS and spatial models. This is particularly true of systems coupling locational analysis capabilities with those of GIS (Densham, 1993a).

A key component of a VIM environment for locational analysis is the user interface. Ideally, this interface supports representations of all four spaces in which decision-makers work: objective, decision, model and geographic. Although a GIS might represent geographic space using a variety of abstractions, the general absence of modelling capabilities from these systems means that they lack representations designed explicitly for objective space, decision space and model space. Furthermore, an interface must support both goal-seeking and "what-if" modes of system use, blending analytical and intuitive approaches to problem-solving. To understand better the issues and problems that must be addressed in designing an interface to represent these spaces, we have designed a model interface (Armstrong, Densham and Lolonis, 1991) for use in redistricting applications. This interface supports bi-directional linkages among graphical displays and analytical capabilities, and direct manipulation of its linked representations.

The interface consists of three windows that display complementary maps, graphs and tables. Used together, these representations provide substantial amounts of information about the four spaces in which decision-makers work. For example, the map window may display current facility locations, and the allocations of demand to them, while the graph window contains a histogram of facility workloads and the table window lists facility attributes. If a decision-maker wishes to investigate the effects of

adding a new facility, two approaches can be used. The first (goal-seeking) approach, is to select a command from a menu that identifies the optimal location for the facility and updates all three windows appropriately. The second (intuitive) approach enables the user to specify the location for the facility by pointing to a candidate location. In this latter case, the map window first must be updated to show existing facilities and all the candidate facility locations that have been identified; the user selects one of these locations and all three windows are then updated to show facility locations and allocations of demand, facility workloads, and facility attributes for the new configuration. In both approaches, the user can confirm the addition of the new facility or reject it, returning the windows to their original state.

Direct manipulation of the interface's representations is used in other contexts. For example, if decision-makers want to investigate the effects of relocating a facility, they may simply click on a facility symbol and drag it to a new location. The allocations of demand are recalculated, redisplayed, and the other windows also are updated to reflect the requested change. In another context, the capacity of a facility may be changed by dragging its bar in the workload histogram to a new level; again, the system automatically recalculates the changes and displays them.

Despite its utility, this interface has shortcomings. An extra window is required to support direct interaction with the underlying analytical operations: the user could set model parameters using either a dialogue box or by manipulating the contents of a graphical display. This window also could be used to stop a model during its solution to change its parameters or to force a particular candidate to be a facility site. This process would be facilitated by animating the solution process, providing a pedagogic tool that would enable a user to see the current best configuration and its attributes.

4.0 INTEGRATING CARTOGRAPHY AND LOCATIONAL ANALYSIS

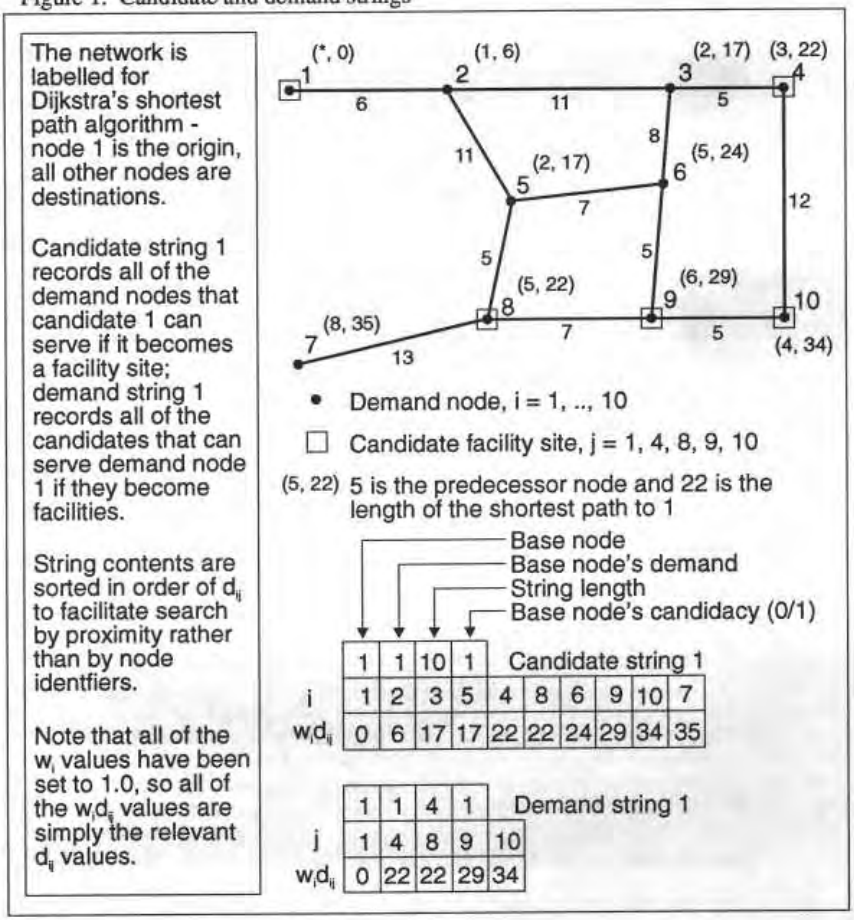
Designers of a visual interactive modelling environment must address issues beyond the design of the interface. First, every operation supported by the interface has implications for the database management system (DBMS) which must provide data for query, analysis and display and must be able to accommodate the results of analyses. Second, data structures must be identified that support both analytical and display operations. Finally, to facilitate human-computer interaction, mechanisms for updating displays in real-time are required.

4.1 Data Structures

Because analytical and display objects often are different in structure, content and function, a central issue in coupling GIS with analytical software is the resolution of object differences between them (Nyerges, 1992). One approach to solving this problem is to employ formal methods for modelling entities and their relationships. Armstrong and Densham (1990) developed Entity-Category-Relationship (ECR) diagrams of two different "user" views of a SDSS database: a cartographic perspective and a locational analysis view. The integration of these two views at a conceptual level required that object differences be resolved. The resulting conceptual design was used with database management software to develop a logical schema and, ultimately, to implement a database.

Several specially-designed relationships and record types are used in the implemented database to minimize search and retrieval to support both analysis and display. To indicate which points in the database are also nodes on the network used for locational analysis, a record called "L_A_Nodes" is used. This record obviates searching all of the points in the database, many of which are not part of the network, to retrieve the network's geometry and topology. To increase the utility of these data after retrieval, relationships are used to indicate the linkages among nodes on the network, sorted in order of network node (L_A_Nodes) identifiers. Sorting these items in the DBMS means that they do not have to be sorted after every retrieval and before they can be processed using a shortest path algorithm. Furthermore, these relationships

Figure 1. Candidate and demand strings



	1	1	10	1	Candidate string 1					
i	1	2	3	5	4	8	6	9	10	7
$w_i d_{ij}$	0	6	17	17	22	22	24	29	34	35

	1	1	4	1	Demand string 1
j	1	4	8	9	10
$w_i d_{ij}$	0	22	22	29	34

also facilitate the identification and retrieval of chains depicting network links for display purposes.

The data structures used to support analytical and display operations are critical in a VIM environment. These data structures must facilitate both goal-seeking and "what-if" forms of analysis, including the generation and evaluation of configurations of facilities and their associated demand sets. Unless these structures also support display, there is a strong possibility that versioning problems will arise - at any point in time, the mathematical representation of the problem is different from the cartographic one. To support the interface described above, and VIM for locational analysis more generally, we are employing data structures developed to support interactive locational analysis.

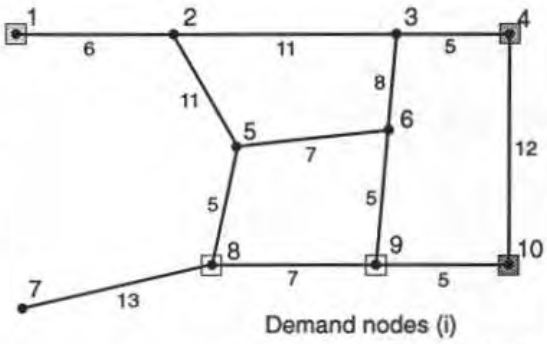
Two types of data structures have been used to develop a series of analytical operations that can be sequenced to implement heuristic location-allocation algorithms. The first structure, the allocation table (Densham and Rushton, 1992a), is a spatial accounting mechanism; it records the closest and second-closest facilities to every demand location. The second structure, the distance string, comes in two forms: candidate strings (Hillsman, 1980) and demand strings (Goodchild and Noronha, 1983). Both types of distance string store information about feasible interactions

Figure 2. The allocation table

Locate facilities at candidates 1, 4 and 10.

The allocation table records the closest and second-closest facility (j) to each demand node (i). The table is built from candidate strings, demand strings, or both.

The objective function value is the sum of the $w_i d_i$ values in the second row of the table.



Demand nodes (i)

		1	2	3	4	5	6	7	8	9	10
Closest facility	j	1	1	4	4	1	10	10	10	10	10
	$w_i d_i$	0	6	5	0	17	10	25	12	5	0
Second-closest facility	j	4	4	10	10	10	4	1	1	4	4
	$w_i d_i$	22	16	17	12	17	13	35	22	17	12

among demand locations and candidate facility locations. Although every demand node has a demand string, only candidate nodes (potential facility locations) have candidate strings.

Figure 1 depicts a small, ten-node network with five candidate locations (nodes 1, 4, 8, 9 and 10) and the associated candidate and demand strings for node 1. The format of the strings is the same: a header row with four fields and two rows with fields containing node identifiers and weighted distances ($w_i d_{ij}$) respectively. The w_i term represents the weight or demand at node i; the "distance" between demand node i and candidate j is represented by d_{ij}. (Here, d_{ij} will be used to denote physical distance but it can be used to represent travel cost, travel time, and other measures of spatial separation.) A demand string records the weighted distances from the demand (base) node to all of the candidates that may serve it if they become facility sites. A candidate string stores the weighted distances to all of the demand nodes that the candidate (base node) may serve if it becomes a facility site. In both types of strings, entries are sorted in ascending order of d_{ij}, the measure of spatial separation. This ordering facilitates search and retrieval by proximity, rather than by node identifiers. Although a complete set of candidate and demand strings contains the same information, they are organized to optimize different forms of data retrieval. Demand strings are used to answer the question "What is the closest facility to this demand node?" In contrast, candidate strings answer the question "Which demand nodes can this candidate serve if it becomes a facility?"

An allocation table consists of six rows and n columns, where n is the number of demand nodes on the network. For each demand node, the first two rows of the table store, respectively, the node identifier of, and the associated weighted distance to, the closest facility; similarly, the third and fourth rows store the identifier of, and the weighted distance to, the second-closest facility. The fifth and sixth rows of the table are used to evaluate alternative locational configurations that are different from those represented in rows one to four. Figure 2 depicts the first four rows of the allocation table for the network introduced in Figure 1 with facilities sited at candidates 1, 4 and 10. An allocation table can be built in two ways: first, by searching along every demand string to find the identifiers of the first two candidates which are facility sites;

and, second, by comparing the weighted distance for each demand node in the facilities' candidate strings.

In concert, an allocation table and a set of candidate and demand strings provide a flexible representation of a locational problem. Densham and Rushton (1992a, 1992b) show how three heuristic location-allocation algorithms can be implemented using five operations defined upon distance strings and the allocation table. A microcomputer-based software package implementing these three algorithms has been developed around these data structures and operations (Densham, 1992). By defining a few additional operations, these data structures also support a further four algorithms (Densham, 1993b).

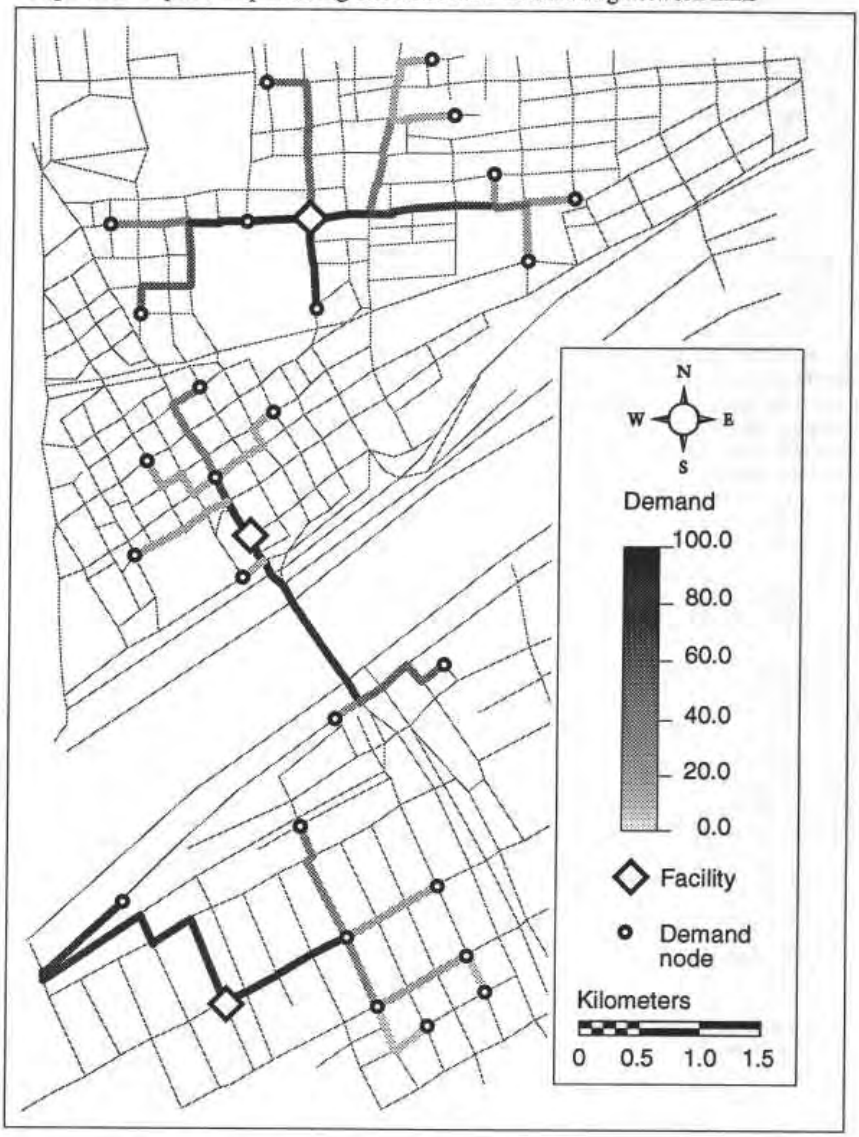
4.2 Generating Spider Maps

We have developed a taxonomy of cartographic displays that can be used to depict the results of different queries and analyses to decision-makers (Armstrong *et al.*, 1992). Each of these maps is suited to answering different questions that decision-makers articluate as they investigate location-selection problems. A "spider map," for example, is used to show the allocation of a dispersed set of demand to a smaller set of facilities; it can be used to depict an existing locational configuration or to show the results of analyses, both goal-seeking and intuitive. A canonical spider map links each demand node to its allocated facility with a straight line. The display of such a map requires geometrical, topological and thematic information: thematic information is used to differentiate among facility locations and demand node locations; geometrical information is used to locate both facilities and demand nodes in space; and, finally, topological information is used to link the demand nodes to the facilities. While distance strings and the allocation table can supply some of this information, the remainder must come from other sources.

The geometry of a network normally is discarded as a shortest path algorithm generates distance strings. This occurs because strings record only those allocations of demand nodes to candidates that are feasible using one or more criteria. Although the spatial separation of demand nodes and candidates is captured in their $w_i d_{ij}$ values, d_{ij} may represent various metrics of spatial separation and the resulting proximity values may bear little relationship to network geometry. Thus, only an abstracted form of the network's topology is retained: pairs of candidates and demand nodes that potentially can fill the roles of service provider and client. Although the allocation table is built from information stored in the strings, it contains two other pieces of information required to generate spider maps. First, the allocation table contains thematic information about which candidates are facility sites and, second, topological information that describes the allocation of demand nodes to these sites. The missing element is the geometrical data - the locations in space of each demand node and facility.

A network's geometry and full topology are required by the shortest path algorithm to build distance strings. In both the PLACE (Goodchild and Noronha, 1983) and LADSS (Densham, 1992) packages, these data are stored in two files: the links file contains the topology, while the geometry of the nodes and their thematic information are stored in the nodes file. The links file contains one record for every node on the network. Each record stores the node's identifier; its valency (the number of network links for which it is a node); the identifiers of the nodes at the other end of its links (sorted in ascending order of identifier); and the corresponding link lengths. The nodes file is a table that contains one, six-field record for every node in the links file. The fields in each record store a node's: identifier; region identifier (a polygon representing a census tract, for example); weight or demand; candidacy (1 if it is a candidate, 0 otherwise); X coordinate; and Y coordinate. The contents of these two files can be retrieved from the database described above. Moreover, because of the organization of these data in the DBMS, they do not have to be sorted before they can be used by the shortest path algorithm (Armstrong and Densham, 1991). Thus, the geometrical data required to generate a vector spider map can be retrieved directly from the database, from the nodes file on disk, or from a data structure in RAM.

Figure 3. A spider map showing allocations and flows along network links



In many situations, a spider map that depicts the actual paths taken through the network enables a decision-maker to understand better the consequences of selecting a particular locational configuration. Figure 3 shows a network-based spider map that depicts the aggregate flows of demand along each link (McKinney, 1991). Producing such a map requires all of the information used for a vector spider map, plus a list of the links that form the paths between each demand node and its closest facility. Although a shortest path algorithm (for example, Dijkstra, 1968) identifies these paths as it builds the distance strings, all the paths underlying a string are discarded once it has been built. Consider, for example, a network of 100 nodes arranged as a ten-by-ten lattice. If all of the nodes have positive weights and also are candidates, then each candidate and demand string has 100 entries. The paths underlying a complete set of

candidate and demand strings for this network traverse 66,000 links, all of which would have to be stored. A 100 node network is tiny when compared with the digital network data sets available from various sources, including TIGER files, and storing paths for such networks would require large amounts of storage. Furthermore, many of these paths will never be required. If we consider locating one facility at a corner of the 100 node lattice, only 100 of the 10,000 paths will be required to generate the spider map - traversing only 900 links.

An alternative to storing the paths is to reconstruct them when they are required. This option is attractive because the amount of computation required to build the paths for a network-based spider map normally is very much less than that required to construct a set of candidate and demand strings. Building a set of candidate and demand strings requires multiple paths to be generated per demand node, one for every candidate site on the network to which the node feasibly can be assigned. The spider map, in contrast, requires the generation of only one path per demand node, depicting the route from the node to its associated facility. The allocation table provides a list of the nodes that are allocated to each facility. To generate the paths linking each demand node to its associated facility, a shortest path algorithm can be used. Because the identifiers of the facilities and of their assigned demand nodes are known, the amount of searching performed by the shortest path algorithm is reduced. This is achieved by searching outward from a facility site only until every demand node allocated to the facility has been reached. Thus, supplementing the distance strings and the allocation table with the contents of the nodes and links files provides all the information required to generate a network-based spider map.

For very large networks, the computation required to build even one candidate or demand string can be considerable. Consequently, we have investigated the use of parallel processing to increase computational throughput and, thereby, decrease the solution times of locational analysis algorithms. We have developed a series of strategies for decomposing spatial algorithms into parallel processes (Armstrong and Densham, 1992; Ding, 1993). These strategies have been used to develop parallel processing versions of Dijkstra's algorithm (Ding *et al.*, 1992; Ding, 1993). Parallel processing can be used to reduce the time required to generate the paths depicted in a network-based spider map. Each facility, its associated list of demand nodes, and the contents of the nodes and links files can be passed to a separate processor. After generating the required paths, each processor can then aggregate the demand flows over each link, returning a list of links traversed and the associated volume of demand. The cartographic chain representing each link can then be retrieved from the database and drawn in an appropriate hue (McKinney, 1991) to depict the demand flow.

4.3 Visual Interactive Modelling

The addition of a new facility to an existing configuration changes the allocations of at least some demand nodes to facilities, requiring the update of a spider map. Using the methods described above, the following course of action occurs if a decision-maker adds a new facility:

1) The decision-maker clicks on a candidate facility site to locate a new facility there.

2) The SDSS responds by adding the new facility to the allocation table and updating its contents.

3) The first row of the allocation table and the contents of the nodes and links files are used to generate the paths through the network that will be depicted in a network-based spider map.

4) The paths, their associated lists of links and demand flows, are used to query the database and to redraw the spider map.

5) The contents of the graph and table windows also are updated.

By applying some knowledge about the spatial structure of a location-allocation model to step 3, it is possible to determine which nodes will be reallocated when the new facility is added (Densham and Rushton, 1992a). This knowledge can be used, first, to generate paths only for these reallocated nodes and, second, to recalculate the aggregate demand flows over affected links, further reducing the amount of computation required to generate the spider map.

A similar process is used when the decision-maker either relocates a facility by dragging it across the screen, or removes a facility from the configuration. The only changes occur in step 3 because a different set of operations are applied to the allocation table and distance strings. These operations determine the effects of the relocation or facility removal on the allocations of demand nodes to facilities and the value of the objective function.

5.0 CONCLUSIONS

We have shown how abstracted topological data structures, used for locational analysis, can be supplemented with geometrical and topological information to produce cartographic displays. Several advantages are realized when this approach to map generation is adopted. First, the same data abstractions are used for analysis and display purposes, obviating versioning problems. Second, the data abstractions can be implemented as objects with both analytical and display methods. Third, a degree of scale independence results because this approach supports multiple representations of networks. Fourth, this approach is suitable for highly interactive problem-solving and decision-making because many of the components of spider maps, and other maps used in locational decision-making, can be generated independently and can be decomposed into parallel processes. Finally, in a pedagogic context, the solution processes to various algorithms can be animated to depict different spatial search strategies.

ACKNOWLEDGEMENTS

Partial support for this paper from the National Science Foundation (SES-90-24278) is acknowledged. This paper is a contribution to Initiative 6 (Spatial Decision Support Systems) of the National Center for Geographic Information and Analysis which is supported by the National Science Foundation (SES-88-10917).

REFERENCES

Armstrong, M.P. and P.J. Densham 1990, Database organization strategies for spatial decision support systems: *International Journal of Geographical Information Systems*, Vol. 4, pp. 3-20.

Armstrong, M.P. and P.J. Densham 1992, Domain decomposition for parallel processing of spatial problems: *Computers, Environment, and Urban Systems*, Vol. 16(6), pp. 497-513.

Armstrong, M.P., P.J. Densham and P. Lolonis 1991, Cartographic visualization and user interfaces in spatial decision support systems: *Proceedings*, GIS/LIS '91, pp. 321-330.

Armstrong, M.P., P.J. Densham, P. Lolonis and G. Rushton 1992, Cartographic displays to support locational decision-making: *Cartography and Geographic Information Systems*, Vol. 19(3), pp. 154-164.

Densham, P.J. 1992, *The Locational Analysis Decision Support System (LADSS)*, NCGIA Software Series S-92-3, NCGIA, Santa Barbara.

Densham, P.J. 1993a, Integrating GIS and spatial modelling: visual interactive modelling and location selection: *Geographical Systems*, Vol. 1(1), in press.

Densham, P.J. 1993b, Modelbase management for heuristic location-allocation algorithms: manuscript available from the author.

Densham, P.J. and M.F. Goodchild 1990, *Spatial Decision Support Systems: Scientific Report for the Specialist Meeting*, NCGIA Technical Report 90-5, NCGIA, Santa Barbara.

Densham, P.J. and G. Rushton 1992a, Strategies for solving large location-allocation problems by heuristic methods: *Environment and Planning A*, Vol. 24, pp. 289-304.

Densham, P.J. and G. Rushton 1992b, A more efficient heuristic for solving large p-median problems: *Papers in Regional Science*, Vol. 71(3), pp. 307-329.

Dijkstra, E. 1959, A note on two problems in connection with graphs: *Numerishe Mathematik*, Vol. 1, pp. 101-118.

Ding, Y. 1993, *Strategies for Parallel Spatial Modelling Using MIMD Approaches*, Unpublished Ph.D. Thesis, Department of Geography, State University of New York at Buffalo, Buffalo.

Ding, Y., P.J. Densham and M.P. Armstrong 1992, Parallel processing for Network analysis: decomposing shortest path algorithms on MIMD computers: *Proceedings*, 5th International Spatial Data Handling Symposium, Vol. 2, pp. 682-691.

Goodchild, M.F. and V. Noronha 1983, *Location-Allocation for Small Computers*, Monograph No. 8, Department of Geography, The University of Iowa, Iowa City.

Hillsman, E.L. 1980, *Heuristic Solutions to Location-Allocation Problems: A Users' Guide to ALLOC IV, V, and VI*, Monograph No. 7, Department of Geography, The University of Iowa, Iowa City.

Hurrion, R.D. 1986, Visual interactive modelling: *European Journal of Operational Research*, Vol. 23, pp. 281-287.

McKinney, J.V. 1991, *Cartographic Representation of Solutions to Location-Allocation Models*, Unpublished Master's Project, Department of Geography, State University of New York at Buffalo, Buffalo.

Nyerges, T. 1992, Coupling GIS and spatial analytic models: *Proceedings*, 5th International Spatial Data Handling Symposium, Vol. 2, pp. 534-543.

Peucker, T.K. and N.R. Chrisman 1975, Cartographic data structures: *The American Cartographer*, Vol. 2(1), pp. 55-89.

Beyond Spatio-temporal Data Models: A Model of GIS as a Technology Embedded in Historical Context

Nicholas R. Chrisman
CHRISMAN@u.washington.edu
Department of Geography DP 10, University of Washington
Seattle, Washington 98195 USA

ABSTRACT

Most of the discussion of time in GIS fits into the general topic of developing a useful model of geographic data. Data models, at their most abstract level, describe objects, relationships, and a system of constraints or axioms. So far, most GIS research posits universal axioms with a strong geometric basis. Time is usually spatialized. Models of GIS should develop to include more than the data, since an operating GIS must be connected to its context in social, economic and administrative life. While time might be reasonably represented as an axis in data space, as a technology, GIS develops in a complex, multi-thread system of events. Understanding the historical nature of the participants in the GIS can help sort out the diversity of data models and the inability to develop common understandings. The logic of historical time, with its multiple threads, provides a rich source of axiomatic structure for a more comprehensive model of GIS. A few starting axioms are offered.

Data Models in GIS

Developing a model for the *data* in a geographic information system has remained a core issue in GIS research for over two decades. Recently, many research leaders in the field of GIS have written papers promoting models of spatial data (including, but not limited to Peuquet, 1988; Frank, 1987; Goodchild, 1987; 1992; Nyerges, 1991a). Due in large part to the reliance on mathematical frameworks and certain basic attitudes towards science, these authors have described abstract models based on universal principles. Most of the work on data models describes a structure of objects and their relationships, using the barest geometric axioms to complete the model (see Codd, 1981 for the connection between a data model and a mathematical theorem; also White, 1984; Corbett, 1979). GIS research must focus more attention on axioms; their origins and their utility. At the moment, the concept of a data model limits its application to the data, not the full understanding of a geographical information system as practiced inside human institutions.

Temporal Data Models

Recent research has devoted attention to incorporating time more fully into the data models of GIS (for example Armstrong, 1988; Barrera and Al-Taha, 1990; Hazelton, 1991). In most of the literature, including my own technical works, time is treated as an axis, a dimension of

measurement similar to the spatial case. Langran and Chrisman (1988) reviewed the difficulties of adding time to geometry and attributes. The basic model of time offered – a consensus of research opinion – was a linear axis with a topology imposed by significant events (see Figure 1).

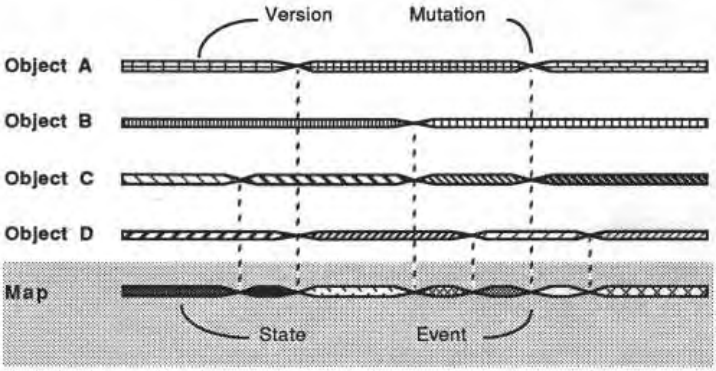


Figure 1: Topology of time (from Langran and Chrisman, 1988)

Langran (1991) provides the fullest discussion to date on the technical issues of incorporating time into GIS. Recent work on data models (such as Pigot and Hazelton, 1992) accept the treatment of time as an axis similar to a spatial one – a concept with deep roots in geography and cartography (Haggett, 1965; Szegö, 1987). As an initial approximation, analytical advances can be developed on this foundation. The use of abstract frames of reference, for both space and time, threatens to ignore the role of other forces in shaping the meaning of the raw measurements. Time in its historical form cannot become an abstract reference, it will be found to defy the unidimensional, monotonic character assigned to it.

Spatio-temporal reasoning should not be presented on a uniform plane against an abstract time axis. The important applications of spatio-temporal reasoning must be historically embedded, involving people in specific locations with complex historical roots to their societies, institutions and cultures. This goal can be achieved by reevaluating the axioms used as a basis for a data model.

Image Schemata: A Path Unlikely to Provide the Axioms

Other research on data models for GIS has recognized the need for some broader context than simply bald coordinate axes. Perhaps the most prominent component of recent literature on geographic data models is a move to add a cognitive component to data models. A number of researchers (including but not limited to: Mark and Frank, 1989; Nyerges, 1991b) have adopted various concepts from the work of Johnson (1987) and Lakoff (1987). While a cognitive emphasis connects the new GIS field more firmly to communication school cartography, this component deflects interest away from redefining the axioms. The concept of "image schemata" places central emphasis on the individual. Johnson's (1987) book is titled *The Body in the Mind*, which leads to mental constructs constrained by the experiential limits of bodies. GIS researchers then postulate that the *relationships* in a GIS must tie to the image schemata of

human body experiences.

Human life and human information has become complex beyond the scope of single bodies. The major developments of the past 10,000 years have not included significant refinement in the human hardware. And yet, medical science has permitted dramatic increases in average longevity, but this achievement is not an example of body-based cognition. Medical research is an example of a social and economic enterprise carried out over the long-term by many cooperating individuals. Most of the interesting advances of the past 10,000 years arise from the creation of larger and larger social and economic structures that endure **despite** the vagaries of individual bodies. Culture is the general covering term for the mechanisms that organize concepts and meanings beyond the scale of the individual. Culture is at least as important to GIS than any image schemata based on individual cognition. The Johnson-Lakoff "image schemata" approach limits attention to the cognition of space inside one body, rather than recognizing the complexity created by cultural transmission of concepts and meaning. The image schemata offers an alternative understanding of relationships but little concerning the axiomatic structure as it is redefined by cultural history.

Discussing a social and cultural element in GIS is no novelty by now. The purpose of this paper is not to repeat the arguments of earlier papers, but to add to their direction. The work on data models has limited the focus precisely to "the data", and the image schemata does not move far into the realm of axioms. So far, there is no satisfactory approach to the historical nature of GIS as a technology and as an element inside human institutions.

VIEWS OF TIME IN GIS

The particular focus of this paper is the treatment of time in combination with geographic information, but not the time that is directly modelled in the database. Inside a data model, an abstract coordinate axis provides a reasonable simplification as a reference for the more complex logic required. However, there are other forms of time as well, particularly the historical sequence of ideas and technologies that lead to some particular GIS in some particular place. Understanding this context is critical to a correct interpretation of the reasons for installing and operating a GIS in some particular manner.

Some of the attitudes about time in GIS are unconscious, as opposed to the highly self-conscious formalisms of current data models. I will present a series of views of technology, each common or applicable in some portion of the field. These views about time and technology are critical to our view of the role of technology in society and thus to our views about GIS research and development.

Unconscious Views of Time

It is very hard to deal with GIS without all the gush of a "new age" or a "revolution" created by the new technology. This view is based on

characterizing the GIS era as signally different from what preceded. At its most extreme, such a view places the past in the murkiness of the Dark Ages with the current era as one of enlightenment (as characterized in Figure 2).

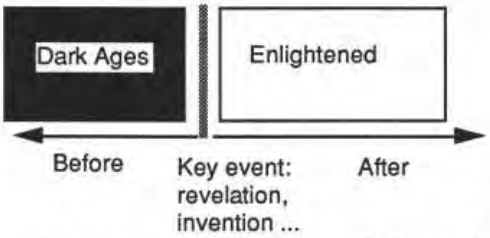


Figure 2: Rudimentary view of history (before and after)

While Figure 2 may seem a caricature, the field of GIS may fall back to this model a bit too easily. The tricky part of this model of development comes after the key event that begins the new age. There seems to be no further need for history, since everything has been resolved. Fukuyama (1992) has recently contended that recent geopolitical changes amount to an "end of history", but those changes simply may represent an end to the cold war mentality of simplifying abstractions; an era of new axioms for international relationships.

Despite the rhetorical recognition of revolutions, the image of time that permeates our culture to a much larger extent is the "March of Progress" (see Figure 3). Time is considered, not just to accompany, but essentially to cause the incremental improvement of material affairs. Since the Enlightenment, the rapid progress of industrial and other technology has contributed to popular acceptance. Time is associated with higher achievement and the phrase "onward and upward" forms an automatic association.

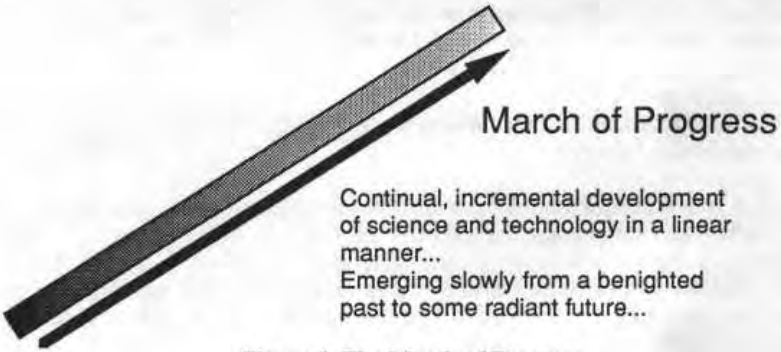


Figure 3: The March of Progress

In GIS, there are echoes of the "March of Progress" theme, particularly in the recent trade literature. The GIS "industry" thrives on the bandwagon effect, the impression that improvement is inexorable and assured. Certain advertisements simply present a list of customers adopting the

particular product, thus fueling the impression that the central issue is simply participating, not the nature of the implementation. The general tone of feature articles in the trade literature trumpets the success (real or imaginary) of each new system. It is just such articles that bring the most furious attacks from the academic community (such as Smith, 1991).

It certainly may be true that some changes seem inevitable. For example, any computer purchased today will seem totally obsolete in a few years, even next month. As the components of the technology continue to get cheaper, do we arrive at the asymptote, the "zero cost hardware"? Even if we do, however, the ideas informing the software and the systems are not guaranteed to progress in any inexorable linear manner. Current adopters of GIS may not be assured of sharing in the success of those who have gone before, unless they listen very closely for the tales of the hard work associated with the successes (such as the memorable "advice" boxes in Huxold and others, 1982). During heady expansion, the distortion of stories, making success seem inevitable, does not serve any purpose. The image of a march of progress is so pervasive that it may be accepted without proof.

Beyond the March of Progress

As a refinement of the March of Progress, a purely linear view of history may be replaced with a recognition of competing forces and ideas. While expressed in various forms, Hegel's dialectic (see Figure 4) captures the basic framework. Marx's (and Engel's) view of history relies on this basic model. Like the march of progress, history follows an essentially linear set of phases or stages. Each stage develops out of the resolution of the previous problems and somehow inevitably causes contradictions that bring about the next conflict. Despite the overall linearity, at any one time the Hegelian dialectic does admit that there will be alternative explanations, thus different interpretations of history.

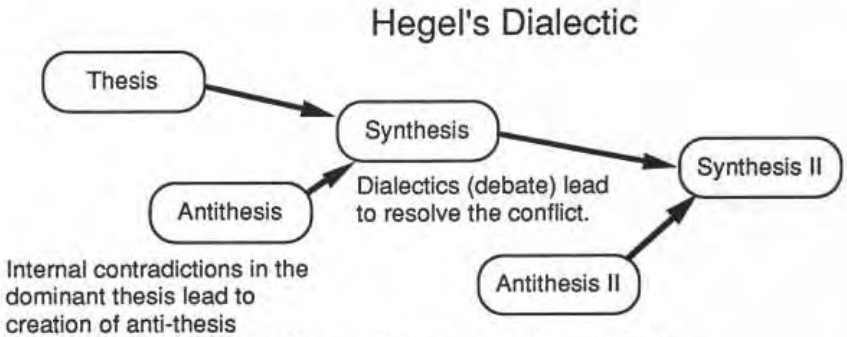


Figure 4: Hegel's dialectic (grossly simplified)

In the field of GIS, there is often the presumption of some basic stages of development, an application of Rostow's paradigm of economic development. There is an assumption that GIS tools all lead in one direction. The differences between raster and vector, and the other technological differences are seen as a part of a universal science

(Goodchild, 1992) that will incorporate the variants under one set of principles. This logic falls into the Hegelian camp.

Reasoning about history is not the same as reasoning about some constructed abstract world with a simple time axis. In historical context, there is no guarantee that the background for each event is unified and coherent. In Usher's (1954, p.19) words, "every event has *its* past." Schematically, this view of history can be presented as multiple "systems of events" or threads of connection that form the historical whole out of parts that may not be inherently consistent (see Figure 5).

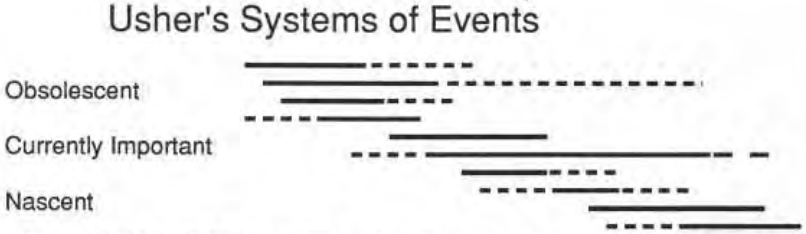


Figure 5: Usher's System of Events (redrawn from Figure 3, Usher, 1954, p. 22)

Each line in Figure 5 can be treated as an orthodox time line with a linear logic. The dashed sections at either end represent the periods in which an idea or construct may be nascent or obsolescent, while the solid portion represents a period of acceptance. Over time, some systems may join as certain concepts become connected. The scheme permits Hegelian synthesis when it actually occurs and recognizes when it may not occur for a long period. Usher's view of historical development could assist in developing the historical elements of GIS.

AN EXAMPLE OF MULTI-THREADED TIME

What does a concept of multi-threaded time have to offer the study of geographic information systems? It provides a framework to understand the ambiguities and conflicts within the technology and the institutions involved. Some of these can be developed by a hypothetical example based on a county land information office, say in the State of Wisconsin. This office has been charged with the "modernization" of the land information in the county, basically a conversion from a preexisting manual system to a computerized GIS. Phrased in these simple terms, it seems to fit the view of progress exemplified by Figure 2 or 3. These models certainly were a part of the political process used to obtain approval in the state legislature and in the county boards. Wisconsin believes in progress. The problem often comes because the vision of progress is different for various participants.

Each discipline or profession can be represented on their own time line (or at least somewhat distinct from their surroundings). In part, these time lines deal with the history of technology as applied by the particular profession. Surveyors have used analytical trigonometry for centuries, so

a computer fits readily into the role of the slide rule and calculator. For other people working for the same county, computers were brought in as accounting machines to print the tax bills, so the model of computing is totally different. These time lines were able to persist so long as the surveyors and accountants work at different ends of the building and their efforts were loosely coordinated. Groups with less computing experience, the cartographers who maintained the manual mapping system using drafting tables and pens and the conservationist (trained to map soils and build farm ponds), might find themselves now more central than they were before, as the GIS emerges to coordinate all the county's records in a spatial framework.

Due to the distinct time lines, each discipline may have distinctive expectations about institutional arrangements. Certain mapping professions, like photogrammetry, have seen a centralized, high capital form of technological innovation over most of the twentieth century. The data processing professionals have similar values to centralization based on mainframe computers. These groups will be ill-prepared for a decentralized workstation network that evolves through collaboration perhaps without a rigid hierarchy.

The time lines discussed above deal with the immediate technological expectations of the participants. This range of time covers basically the past century at most and often remains within the work experience of the participants. This is no the only scale of time involved in this county modernization effort. Beyond the history of the technology, the county land information offices connect to a series of historical threads that define the fundamental meanings of their functions.

Many of the high–priority elements of the land information plan deal with environmental regulation, to preserve wetlands, reduce soil erosion, and many other specific programs. These efforts fit into a time line of the environmental movement – a relatively short term affair, though Aldo Leopold, the local saint, wrote his books in the early part of this century. Along this time line, Ian McHarg's *Design with Nature* (1969) creates a close linkage to the time line of the GIS technology. The message of integrating various forms of information was a part of this current of thought before the computer was applied to it.

The time lines of other components in the county land information puzzle are much deeper seated and much less clearly articulated by the participants. The legal structure of the land records is a long and complex history tracing back to Roman concepts of persons and rights, plus medieval constructs of common law – precedents and rules of evidence. These legal schemes came into a wide diversity of form in Europe, and certain elements were brought to the colonies (see Cronon, 1983 and Fischer, 1989). Parts of Wisconsin have the geometric forms of French long lots as signs of the diverse origins (Fortin, 1985). The particular elements borrowed from Britain and France may have disappeared in their original form, due to later events in the history of those nations. The american version of the Enlightenment, with its restoration of certain Roman forms, had a deep influence on the landscape of the

American West. Uniform squares were placed on the landscape far beyond the most grandiose legionary concession. But the time reference is not just to a rationalist vision. Surveyors are enjoined to "follow in the footsteps" of the original survey for ever more. The primitive surveying technology of the 1820's will continue to influence our Wisconsin county.

The Public Land Survey represents an amazing vision of uniformity and rationalism that conflicts with Leopold and McHarg's visions of connection to the particularities of the ecological system. The conflict between a geometric view and an ecological systems view are not resolved by having each one enshrined in statute and ordinances (Sullivan and others, 1985). These differences are not simply about GIS technology, though they will create different views about the capabilities required for that technology. The legal structure of property rights involves conflicting time lines. The basic constitutional structure is built on Locke's theories of government, with its assertions of individual rights. Modern environmental regulations restrict those rights without completely instituting the requisite changes to alternative views of rights and duties (Bromley, 1991). In the matter of wetlands regulations, the time lines stretch back to the medieval common law treatment of water rights, now reinterpreted in a totally different circumstance.

Most of these historical connections might seem remote to the participants, but the origins of their beliefs and values arise from these time lines. The diversity of different traces of thinking will have practical consequences in the implementation and management in the prototypical county in Wisconsin and anywhere else. The set of historical threads will vary, but there will undoubtably be many of them. What seems like inevitable progress to one participant may seem totally incompatible to another. Resolving these conflicts may be solved through the methods focused on individual perception and behavior, but that may only resolve the conflict for the specific participants, not their whole support network of professional or disciplinary background. A long term solution will require confronting the different views of history.

BUILDING A SET OF HISTORICAL AXIOMS

Beginning with Hegel's dialectic and continuing with the Wisconsin example, it should be apparent that sometimes conflicting ideas can coexist. In fact, despite Hegel's grand assertions of dialectical certainty, conflicting ideas can persist over quite long periods. Just because two events occur simultaneously, one cannot infer that the two events are connected in any direct way, or that their resolution is simply a matter of time. Relationships through time do not become instantly consistent. Similarly, some particular element may persist long after the initial reasons for its creation have faded away.

> "(M)any systems have persisted from a remote past, and have lost all significant contact with current patterns of behavior. ... Hunting privileges, for instance, were among the most useless and most irritating survivals of feudalism. The duty of protecting the peasantry and their crops from wild animals degenerated into a

right to preserve game and to hunt over the peasant's crops, to the detriment of every vital interest of the peasant." (Usher, 1954, p. 23)

Another concern in temporal reasoning involves causation. One of the classic flaws in logic is to assert that prior events automatically cause later ones. The flaw is known as *post hoc– propter hoc*. More connection is needed than simple prior occurrence.

To summarize in more succinct terms:

Axiom 1 (multiple time lines): Simultaneity is no guarantee of connection.

Axiom 2 (post hoc propter hoc): A prior event may be totally unrelated to (and uninvolved in the causation of) a later event.

Axiom 3 (inevitable conflict): Since each group in society and each institution has a distinct historical development of their beliefs and values, distinct and incompatible definitions of objects and relationships must be expected.

Axiom 4 (situated definition): Since definitions of legal (cultural and social) rights and responsibilities vary among societies and develop over time, the design of a GIS must be seen as conditional on the context for which it is designed.

CONCLUSION

A model of GIS cannot treat time as a sterile, abstract dimension without losing the historical specificity of its context. Modelling of GIS must extend beyond the data stored in the GIS to include the institutions that adopt the technology and the conversions of the industry that manages spatial information. Once these components are included, a model of time must include specific events and historical processes that lead to meanings of geographic phenomena amongst the diverse participants.

The field of GIS is beset by cultural expectations about time and progress. These cloud the importance of historical context in the implementation and development of these technological changes. Research in GIS should not stop with models of the data inside the GIS. The technological development process itself has historical roots and progresses in paths not entirely picked for the purest reasons. Even more importantly, the meaning of that data comes from the users and from the context of the uses. The context in turn is heavily influenced by the historical processes that lead to this point. A multi-thread model of historical origins seems much more appropriate than the single axis models that pervade the current thinking.

The purpose of GIS remains to integrate information from diverse sources. At the start, simple tricks such a spatial collocation (polygon overlay) could suffice. These tools remain valid if applied within a well-recognized and accepted framework of axioms. What must be designed is a method to integrate totally opposing frames of reference. This will not be carried out simply by mapping objects and relationships between two schema, because the axiomatic structure is the fundamental issue.

References Cited

Armstrong, M. 1988: Temporality in spatial databases. *Proceedings GIS/LIS 88*, 880-889.

Barrera, R. and Al-Taha, K. 1990: Models in temporal knowledge representation. *NCGIA Technical Report 90-8*. University of Maine, Orono.

Bromley, D.W. 1991: *Environment and Economy*. Blackwell, Cambridge MA.

Chrisman, N. R. 1987: Design of geographic information systems based on social and cultural goals. *Photogrammetric Engineering and Remote Sensing* **53**: 1367-1370.

Codd, E.F. 1981: Data models in database management. *SIGMOD Record*, **11**: 112-114.

Corbett, James 1979: Topological Principles in Cartography, *Research Paper 48*. US Census Bureau: Washington DC.

Cronon, William. 1983: *Changes in the Land: Indians, Colonists and the Ecology of New England*. Hill and Wang: New York.

Fischer, David H. 1989: *Albion's Seed: Four British Folkways in America*. Oxford Press: New York.

Fortin, Lucie 1985: *The Evolution and Continuance of Contrasting Land Division Systems in the De Pere Region of Wisconsin*. unpublished MS thesis. University of Wisconsin–Madison.

Frank, Andrew U. 1987: Towards a spatial theory. *Proceedings International GIS Symposium: The Research Agenda* **2**:215-227.

Fukuyama, Francis. 1992: *The End of History and the Last Man*. Free Press, New York.

Goodchild, M.F. 1987: A spatial analytical perspective on geographical information systems. *International Journal of Geographical Information Systems*. **1**(4): 327-334..

Goodchild, M.F. 1992: Geographical information science. *International Journal of Geographical Information Systems*. **6**(1):31-45.

Haggett, Peter 1965: *Locational Analysis in Human Geography*. Arnold, London.

Hazelton, B. 1991: *Integrating Time, Dynamic Modelling and GIS: Developing a Four-Dimensional GIS*. PhD dissertation, University of Melbourne.

Huxold, W.E. Allen, R,K. and Gschwind, R.A. 1982: An evaluation of the City of Milwaukee automated geographic information and cartographic system in retrospect. paper presented at Harvard Graphics Week 1982.

Johnson, M. 1987: *The Body in the Mind: The Bodily Basis of Meaning, Imagination and Reasoning*. U. Chicago Press, Chicago.

Lakoff, G. 1987: *Women, Fire and Dangerous Things*, U. Chicago Press, Chicago.

Langran G. 1991: *Time in Geographic Information Systems*. Taylor & Francis, London.

Langran, G. and Chrisman, N.R., 1988: A Framework for Temporal Geographic Information, *Cartographica*, **25**(3): 1-14.

Mark, D.M. and Frank, A.U. 1989: Concepts of space and spatial language. *Proceedings Auto-Carto 9*, 538-556.

McHarg, Ian 1969: *Design with Nature*. Doubleday, Garden City NY.

Nyerges, T.L. 1991a: Analytical map use. *Cartography and Geographic Information Systems*. **18**: 11-22.

Nyerges, T.L. 1991b: Geographic information abstractions: conceptual clarity for geographic modeling. *Environment and Planning A*. **23**:1483-1499.

Peuquet, Donna J. 1988: Representations of space: Towards a conceptual synthesis. *Annals AAG*, **78**: 375-394.

Pigot, S. and Hazelton, B. 1992: The fundamentals of a topological model for a four-dimensional GIS, *Proceedings 5th International Symposium on Spatial Data Handling*, **2**: 580-591.

Smith, Neil. 1992: History and philosophy of geography: real wars, theory wars. *Progress in Human Geography*. **16**(2):257-271.

Sullivan, J.G., Chrisman, N.R. and Niemann, B.J. 1985: Wastelands versus wetlands in Westport, Wisconsin: Landscape planning and tax assessment equalization. *Proceedings URISA*, **1**:73-85.

Szegö, Janos. 1987: *Human Cartography: Mapping the World of Man*. Swedish Council for Building Research: Stockholm.

Usher, Abbott P. 1954: *A History of Mechanical Inventions*. second edition, Harvard Press, Cambridge MA.

White, M. 1984: Technical requirements and standards for a multipurpose geographic data system, *The American Cartographer*, **11**, 15-26.

PARTICIPATORY MULTI-OBJECTIVE DECISION-MAKING IN GIS

J. Ronald Eastman, James Toledano, Weigen Jin, Peter A.K. Kyem

The Clark Labs for Cartographic Technology and Geographic Analysis*
Clark University
Worcester, MA 01610, USA.

ABSTRACT

To be effective in a broad range of decision making contexts, GIS procedures for multi-objective decision making need to be *participatory* in nature -- where participatory implies that the decision maker plays an active role in the decision process and finds the procedures used to have a simple intuitive understanding. In this paper we concentrate on the development of such a procedure for multi-objective decision making in cases with conflicting objectives. The procedure is based on a choice heuristic that is simple to understand and is capable of rapidly processing the massive data sets of raster GIS. Details of the algorithm are presented along with an experimental assessment of alternative procedures for the resolution of conflict cases.

INTRODUCTION

As an analytical technology, GIS offers to environmental management the opportunity for a more explicitly reasoned decision making process, particularly in cases where multiple objectives and criteria are involved. However, the field has been slow to adopt explicit decision support procedures, most commonly relying upon Mathematical Programming solutions external to the GIS (e.g., Diamond and Wright, 1988; Janssen and Rietveld, 1990; Carver, 1991; Campbell et al, 1992). While such procedures may be effective in certain contexts, they are unworkable in the context of raster GIS (because of data volume), require software often unavailable to most GIS users, and are intimidating to the majority of decision makers. As a result, the Clark Labs have been involved, over the past year and a half, in the development of ***participatory*** GIS decision making procedures.

PARTICIPATORY TECHNIQUES

The need for participatory techniques arises most particularly from the recognition that decisions happen over a broad spectrum of management levels, often requiring the input from groups of affected persons. For example, a government organization might enact a directive that will allow small land owners the opportunity to plant certain cash crops once only permitted to large estate holders. This is a *policy decision*, made by individuals who perhaps quite nicely fit our traditional concept of decision makers. However, the decision of whether to actually plant a particular crop on a given piece of land (a process

* The Clark Labs for Cartographic Technology and Geographic Analysis comprise one of four Centers within the George Perkins Marsh Institute of the Graduate School of Geography at Clark University. The Clark Labs produce and distribute on a nonprofit basis the IDRISI Geographic Information System software package.

be called a *resource allocation decision*) is left to the community members. Thus decisions that affect the environment can take place at a variety of levels, from national to regional, to district and community levels -- even to that of individual farms!

In this example, both groups of decision makers could benefit from GIS. At the policy level, the system would typically be used to provide contextual information. Very rarely, it might be used to predictively simulate the decision behavior of relevant stakeholders so that the potential impacts of policy decisions can be evaluated. At the resource allocation level, individual communities could benefit from an examination of the many criteria that affect the suitability of land for a particular use. Ideally, the GIS would also be used to weight the differing criteria associated with any particular objective and to allow comparative evaluation of differing objectives that compete for access to the land.

All of these scenarios, however, are unlikely in the current climate of GIS. The technology requires a broad background in areas as diverse as statistics, geographic modelling, geodesy, cartography, remote sensing, photogrammetry and the like, and requires demonstrated skills in computer use and the extensive suite of software tools involved in the typical GIS. In addition, being new and very dramatic in its presentational quality, it is commonly wrapped in a mystique of high science and overstated expectations. To add to this the highly mathematical nature of techniques such as linear programming only compounds the problem. As a result, these are hardly the circumstances suitable for involving the majority of national and regional level decision makers, let alone individual farmers!

From this we have concluded the need for *participatory* techniques that can involve the broad spectrum of potential stakeholders in the decision making process. At the outset we see a different role for the GIS analyst -- not as an expert possessor of the new science, but as a facilitator, much like a focus group leader, that can work with groups and individuals to make the system work for them. However, this role is unlikely to succeed without the development of new analytical procedures that are :

1. suitable for use with groups of stakeholders as well as individuals. To do so, they must explicitly incorporate procedures for the development of consensus and the active participation of all. In addition, they must have;

2. an immediate intuitive appeal. The need for a rigorous mathematical basis is important. However, without a strongly intuitive character they are unlikely to gain the confidence and support of the decision makers involved.

The first of these issues has been addressed in previous papers (e.g., Eastman et al., 1992; Eastman et al., 1993a; Eastman et al., 1993b), particularly the use of group interactive techniques for the development of weights in Multi-Criteria Evaluation. In this paper, however, we address the issue of decision procedures with strong intuitive appeal, with particular reference to the problem of multi-objective decision making under conditions of conflicting objectives.

MULTI-OBJECTIVE DECISION MAKING

In the context used here, an *objective* is understood to be a perspective or philosophy that guides the structuring of a decision rule (see Eastman et al., 1993a and Eastman et al., 1993b for a more extensive treatment of terms and definitions). Thus a multi-objective

problem is one in which several different perspectives are drawn upon to determine the allocation of a particular resource. In some cases multiple objectives are complementary, as in multiple-use land allocations. However, very frequently, objectives are in conflict and must be resolved either by means of prioritization or by conflict resolution. Our concern here is with the latter.

In the procedure developed here, each of the objectives is first dealt with as an independent multi-criteria evaluation problem. The results of these single objective solutions are then used to determine areas of conflicting claims in need of a compromise solution.

SINGLE OBJECTIVE MULTI-CRITERIA EVALUATION*

The first step in the single objective evaluations is to identify the relevant criteria that will allow, for each piece of land, a determination of suitability for that objective. These criteria may be either in the form of continuous factors (such as slope gradients) or boolean constraints (such as protective buffers). Continuous factors are then scaled to a standardized range and a set of weights are developed to express their relative importance to the objective under consideration. Here a group participation technique is used whereby each pairwise combination of factors is rated for relative importance using a simple 9-point relative rating scale (see Eastman et al., 1993b). The principal eigenvector of this pairwise comparison matrix is then calculated (using the WEIGHT module in IDRISI) to develop a set of best-fit weights that sum to 1.0 (Saaty, 1977). The criteria are then combined by means of a weighted linear combination and subsequently masked by each of the boolean constraints in turn (using the MCE module in IDRISI). The result is a continuous suitability map with the same numeric range as the standardized factor maps.

Once the suitability map has been produced, the final issue concerns which cells to choose in order to meet a specific areal goal (e.g., the best 1500 hectares of land for that objective). In the context of traditional approaches to land allocation, this problem would typically be expressed as the need to determine that set of cells in which the sum of suitabilities is maximized, subject to the constraint that the total area of the included cells match a specific goal. This is known variously as an *objective function* or *performance index* (see Diamond and Wright, 1989). In this discussion it will be called a *choice function*, to distinguish it from its counterpart, a *choice heuristic*. With a choice heuristic, rather than define a mathematical basis for selecting the optimum set, a procedure is set out that accomplishes the same result, or is its close approximation. From a participatory perspective, choice heuristics are desirable because of their simplicity of understanding and ease of application.

In the case being considered here, selecting the best x hectares of land for a particular objective, using a choice function, is extremely tedious. In theory, every possible combination of cells would need to be tested to evaluate which would maximize the sum of suitabilities. Most procedures that use choice functions use techniques to reduce the number of examined cases to only those that are likely candidates (such as the Simplex Method in Linear Programming). Regardless, the procedure is calculation intensive and essentially impossible with the large data volumes of raster GIS. However, a simple choice heuristic can accomplish the same result with a minimum of work. By simply rank ordering the cells in the suitability map and taking the best ranked cells to meet the area target, the same result is achieved -- the sum of the suitabilities will be the maximum possible.

---

* See Eastman et al., 1993a and 1993b for more detailed discussions of this process.

Rank ordering the enormous volume of data in a raster image is not trivial! Standard automated procedures such as a Bubble, Shell or Quick Sort simply cannot handle the problem in sufficient time. Accordingly, a simpler procedure was used in the development of the RANK module for the IDRISI system. By specifying that the suitability map must be in the form of an 8-bit integer (i.e., byte) image, only values from 0-255 can occur. This was felt to give adequate precision to the suitability map and allowed a rapid procedure for sorting based on the allocation of image cells to a 256-bin counting array. RANK also allows the sorting of ties by reference to a second image. Thus the counting array consists of a 256 x 256 bin structure. Sorting in this fashion is extremely rapid, with the rank image being output as a real number floating point image to accommodate the large numbers that can result.

Once the suitabilities have been ranked, it is a simple matter to select a set consisting of the best cells for a particular area goal. If, for example, the best 5000 cells were required, the rank map would be reclassified into a Boolean map of the result, by assigning a 1 to the ranks from 1 to 5000 (assuming descending ranks) and a 0 to all that are higher.

THE MULTI-OBJECTIVE PROCEDURE

Given the basic logic of the RANK/RECLASS procedure for allocating land based on a single objective, we developed a multi-dimensional extension named MOLA (for *Multi-Objective Land Allocation*) for extending a similar logic to cases with conflicting objectives.

The Basic Allocation Procedure

In teaching the MOLA procedure, we have commonly started by producing single-objective allocation maps for a two-objective problem and then overlaying the results to see how they conflict. Doing so (using the CROSSTAB procedure in IDRISI) yields four classes:

1. areas selected by Objective 1 and not by Objective 2
2. areas selected by Objective 2 and not by Objective 1
3. areas not selected by either objective, and
4. areas selected by both objectives (and thus in conflict).

Clearly it is the conflict areas that need to be resolved. This can also be understood by means of a decision space established by using the suitability scale for each objective as a separate axis in a multi-dimension space (Figure 1). Each cell in the region can be located in this space based upon its suitability on each objective. The RANK/RECLASS procedure discussed earlier is thus equivalent to moving a perpendicular decision line down from the position of maximum suitability until enough cells are captured to make up the areal goal. In the case of two objectives, the two decision lines clearly delineate the four regions just discussed (Figure 1). The basic underlying procedure of MOLA is thus a reclassification of ranked suitability maps with a subsequent resolution of conflicting cells.

Resolution of Conflicts

Clearly the simplest procedure for the resolution of conflicts would be to split the conflicting cells evenly between the objectives. A simple heuristic for doing this would be to

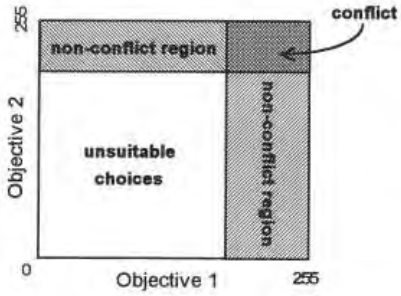


Figure 1

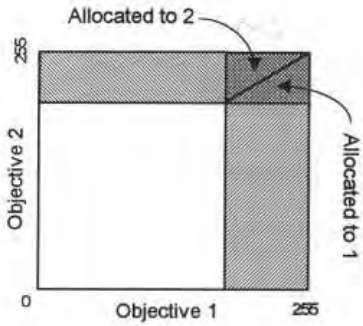


Figure 2

draw a diagonal across the conflict zone and thereby allocate cells to that objective for which it is marginally best suited (Figure 2). However, we have found that a better procedure is to allocate conflicting cells based on a single decision line that bisects the entire decision space (Figure 3). To the extent that this line divides the conflict zone, the proportional division of cells will be determined (Figure 4).

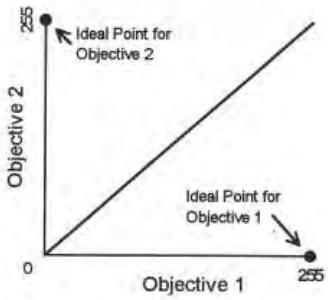


Figure 3

The logic of using a single decision line comes from a simple consideration of the problem at hand. With a conflict cell, a decision needs to be made about which of the objectives it should be allocated to. Logically, it would seem that this should be that objective for which the cell is most inherently suited. With conflicting objectives, each can be considered to have an *ideal point* defining the case of a cell that is maximally suited for that objective and minimally suited for all others (Figure 3). This single line (actually a hyperplane in multidimensional space) thus defines the best allocation for each cell based on a *minimum-distance-to-ideal-point* logic.

If one now considers how this single decision line differs in its resolution of conflicts from the conflict zone diagonal approach (Figure 5), it is clear that in the latter case, suboptimal allocations might be made. Whenever the areas to be allocated to each objective differ considerably, the conflict zone will form an elongated rectangle that is displaced from the *minimum-distance-to-ideal-point* line. As a result, some cells that are better suited for one objective will be allocated to the other (the shaded zone in Figure 5).

This effect is quite dramatic and can lead to substantially counterintuitive allocations. As a consequence we have adopted the use of the single *minimum-distance-to-ideal-point* decision line in the MOLA procedure. However, there are four further reasons for adopting this approach. First, this single *minimum-distance-to-ideal-point* decision line provides a simple logic for the weighting of objectives. Using the case of two objectives for illustration, a 45 degree line implies equal weight being assigned to the two objectives in the allocation of conflict cases. Changing this relative weight requires no more than simply

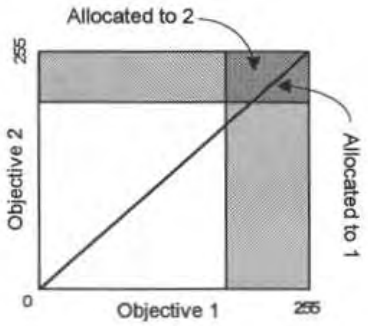


Figure 4

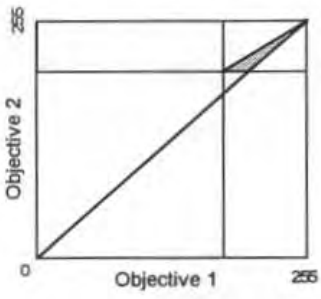


Figure 5

changing the angle of the line. In fact, the ratio of the relative weights assigned to those objectives defines the tangent of that angle. In practice, however, assignments are made by the equivalent process of comparing the magnitudes of the weighted suitabilities. Second, a 45 degree line will maintain weighting consistency during the iteration process (with the simple diagonal, the weight will vary as the rectangle changes in form).

The other two reasons supporting the use of a single *minimum-distance-to-ideal-point* decision line relate to the ability of the procedure to achieve an exact solution and the speed with which it is able to converge. These will be dealt with in the following two sections.

Iteration

Having divided the conflict cells between the objectives, it is clear that both will be short on meeting their area targets. As a result, the MOLA procedure then calculates the deficit for each objective and moves the decision lines further down into the zone of poorer choices. This procedure continues until either an exact solution is achieved or it is clear that the procedure cannot converge on a solution. From experience with the two procedures for resolving conflicts discussed above, it has been noticed that the simple conflict zone diagonal approach quite commonly has difficulty in reaching an exact solution, while the single *minimum-distance-to-ideal-point* decision line rarely does. It would appear that this results from the allocation of sub-optimal cells. In fact, the simple diagonal bisection approach gives weight in the resolution of conflicts that is directly proportional to the size of areal targets. Thus an objective with only modest areal goals will have difficulty in competing for conflict cells with another that is destined for a much larger allocation.

To illustrate this problem, consider the case of a multi-objective decision looking to allocate 1500 hectares to be zoned for industrial uses and 6000 hectares for agriculture*. Figure 6a illustrates the suitability map for industry developed from a multi-criteria evaluation using factors of proximity to water, proximity to roads, proximity to power lines, slope gradient and proximity to market. Figure 6b shows the suitability map for agriculture based on soil capability, proximity to water, proximity to roads, proximity to market, and slope gradient. Figure 6c shows the result of applying the MOLA procedure in IDRISI

* This case study was conducted for the Kathmandu Valley area of Nepal. The industry under consideration was the production of hand-woven carpets. More details about this study can be found in Eastman et al., 1993a and 1993b.

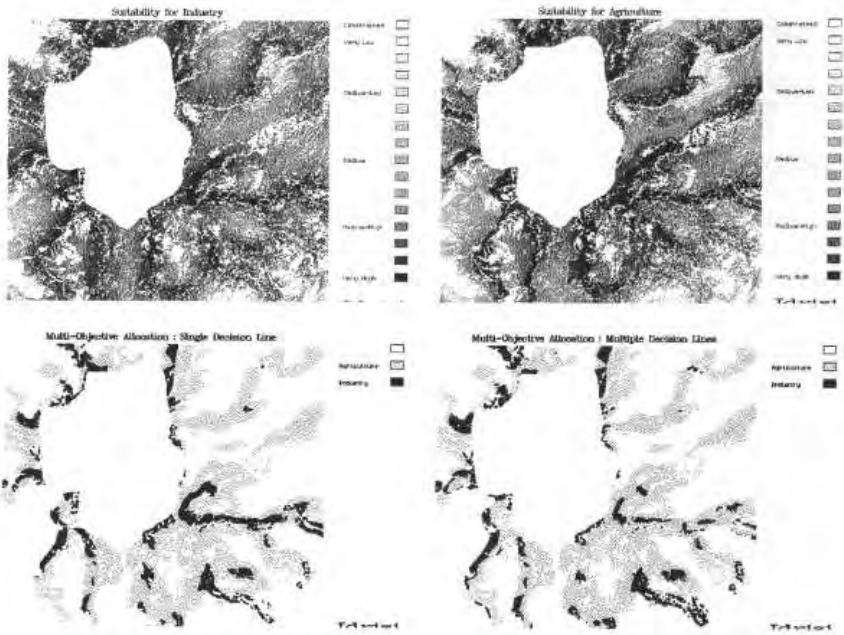


Figure 6 : The results of a Multi-Objective allocation using the two procedures discussed for conflict resolution. Figure 6a (top left) shows the suitability map for industry developed using the MCE module, while Figure 6b (top right) shows the suitability map for agriculture. Figure 6c (bottom left) shows the result from the MOLA module using the single minimum-distance-to-ideal-point decision line. Figure 6d (bottom right) shows the result using multiple decision lines that bisect the conflict zone.

using its single *minimum-distance-to-ideal-point* decision line and equal weight between the objectives. Figure 6d shows the result using a modified version of this procedure that uses a bisection of the conflict zone, leading to multiple decision lines, again with equal weight assigned to the objectives. Table 1 provides additional summary data about these results.

	Rank Goal	Maximum Rank Achieved		% Difference
		Single Decision Line	Multiple Decision Lines	
Industry	16,666	22,818	35,265	55
Agriculture	66,666	80,638	76,960	-5
Iterations		11	19	73

Table 1: Comparison of Multi-Objective Allocations using the Single and Multiple Line Conflict Resolution Techniques.

In Table 1 the *Rank Goal* specifies the lowest (i.e. maximum) rank that would exist among the cells assigned to that objective if it were successful in winning all cases of conflict resolution. The *Maximum Rank Achieved* indicates the worst rank that does occur after the multi-objective procedure. In essence, this indicate how far the horizontal and vertical decision lines in Figures 2 and 5 had to be moved to capture the required number of cells after iterative solution to conflict resolution. As can be seen in Table 1, the use of the multiple decision line procedure had little effect on agriculture (in fact its worst rank improved by 5%) but had a major effect on industry (where the maximum rank got worse by 55%). As can clearly be seen, the larger land claim by agriculture (a four-fold difference) gave it a disproportionate weight in the conflict resolution procedure using the multiple line procedure. Clearly, the single decision line procedure as used in the MOLA module provides a much more even division of land.

Standardization

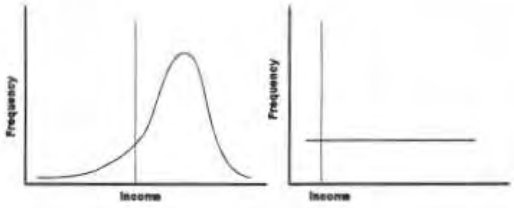
The procedures above describe the complete working of the MOLA routine. However, a vitally important issue is that of standardization. The suitability maps developed through multi-criteria evaluation (such as with the MCE procedure in IDRISI) are developed in isolation without any broader concern for meaning relative to other objectives. Thus there is no inherent reason to believe that a suitability rating of 100 on one objective means the same as a rating of 100 on another.

One obvious procedure for standardization would be to transform the suitability maps to standard scores by subtracting the means and dividing by their standard deviations. A module named STANDARD was created in IDRISI to facilitate this operation. However, the results were rarely satisfactory since the underlying distributions were seldom normal. As a consequence, a non-parametric procedure of standardization was sought.

Considering the case of converting images to standard scores, the underlying logic is one of matching the two distributions to be compared. Another possibility that would meet the non-parametric criterion would be to undertake a histogram equalization of the suitability maps before comparison with MOLA. To illustrate the concept, consider Figure 7a. Here we see the income distribution of two hypothetical countries -- one in which the majority of people have low incomes and the other in which most have high incomes. Comparing two persons from these two countries with equal income it is clear that their level of affluence is unlikely to be the same. In the case of the rich country, that person is likely to be considered fairly poor and would find that their income did not buy a great deal of goods and services. However, the same income in a poorer country would probably be considered to be quite substantial with a corresponding increase in affluence.

With histogram equalization the data values are monotonically changed such that a uniform distribution is achieved. As can be seen in Figure 7b, this results in a substantial repositioning of the two individuals considered above. Now they are no longer considered as equals since the equalization of the poorer country distribution required a net movement of most values to the right while that for the richer country required a net movement to the left.

The simplest means of histogram equalization is simple rank ordering! As a consequence, the input requirement for the basic allocation stage using the RANK/RECLASS decision heuristic is also that which will lead to a non-parametric equalization of the distributions. In effect, histogram equalization gives each cell a value that is relative to the



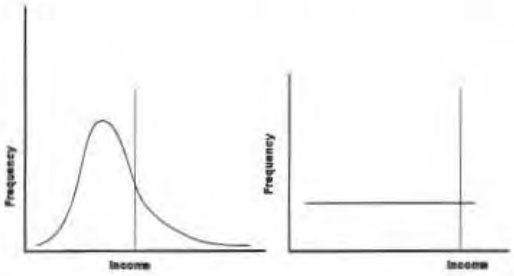


Figure 7a **Figure 7b**

entire group. Thus comparing a value of 100 on a ranked scale for one objective does have a similar meaning to a value of 100 on another.

One of the consequences of rank ordering suitabilities before comparison is that cells will tend to line up along the 45 degree diagonal in decision space -- objectives are commonly correlated to some degree. As a result, the single *minimum-distance-to-ideal-point* decision line will have a tendency to bisect the scatter of cells in decision space. Thus this single decision line will tend to more equitably divide the group in any conflict resolution. We have noticed that the single *minimum-distance-to-ideal-point* decision line technique for conflict resolution tends to solve substantially faster than the simple conflict region diagonal approach. For example, in Table 1 it can be seen that multiple decision line technique took almost twice as many iterations as the single *minimum-distance-to-ideal-point* decision line technique. We believe that to some extent this is because the *minimum-distance-to-ideal-point* decision line is more advantageously situated relative to the main pool of cells. However, it is also clearly related to the increasing tendency of the multiple-line technique to undertake inappropriate conflict resolutions as the difference in areal claims increases. For example, with a 10-fold difference in areal claims (as opposed to the four-fold difference illustrated here), the percentage increase in iterations changes from 73% to 600%!

DISCUSSION AND CONCLUSIONS

In our work with United Nations Institute for Training and Research (UNITAR) and the United Nations Environment Programme Global Resources Information Database (UNEP/GRID) we have had the opportunity to test the MOLA procedure with decision

makers in Nepal, Chile, and a variety of the Baltic nations. Participants have uniformly found the Multi-Objective MOLA procedure to be very simply understood and to be a useful complement to the other participatory techniques, such as the MCE and WEIGHT modules used for Multi-Criteria Evaluation. It is clear from these results that the single *minimum-distance-to-ideal point* conflict resolution procedure is the superior choice over the multiple line technique also discussed -- it avoids the problem of unintended differential weighting based on the relative size of areal claims. Furthermore, in addition to being simply understood, it is fast and capable of processing the large data sets associated with raster GIS -- strong requirements of a procedure that can effectively operate in the interactive context of participatory decision making.

REFERENCES

Campbell, J.C., Radke, J., Gless, J.T., Wirtshafter, R.M., 1992, An application of linear programming and geographic information systems: Cropland allocation. Antigua. *Environment and Planning A* 24, 535-549.

Carver, S.J., 1991, Integrating multi-criteria evaluation with geographical information systems. *International Journal of Geographical Information Systems* 5(3), 321-339.

Diamond, J.T., and J.R. Wright, 1988, Design of an integrated spatial information system for multiobjective land-use planning. *Environment and Planning B: Planning and Design* 15, 205-214.

Diamond, J.T., and J.R. Wright, 1989, Efficient land allocation. *Journal of Urban Planning and Development* 115(2), 81-96.

Eastman, J.R., Jin, W., Kyem, P.A.K., Toledano, J., (1992) Participatory Procedures for Multi-Criteria Evaluation in GIS, *Proceedings, Chinese Professionals in GIS '92*, [in press].

Eastman, J.R., Kyem, P.A.K., Toledano, J., Jin, W., (1993a) *GIS and Decision Making* (Geneva: UNITAR).

Eastman, J.R., Kyem, P.A.K., Toledano, J., (1993b) A Procedure for Multi-Objective Decision Making in GIS Under Conditions of Competing Objectives, *Proceedings, EGIS'93*, 438-447.

Janssen, R., and P. Rietveld, 1990, Multicriteria analysis and geographical information systems: An application to agricultural landuse in the Netherlands. In *Geographical Information Systems for Urban and Regional Planning*, eds., H.J. Scholten and J.C.H. Stillwell (Amsterdam: Kluwer Academic Publishers) 129-139.

Rosenthal, R.E., (1985) Concepts, Theory and Techniques: Principals of Multiobjective Optimization. *Decision Sciences*, Vol. 16, No. 2, pp. 133-152.

Saaty, T.L., (1977) A Scaling Method for Priorities in Hierarchical Structures, *J. Math. Psychology*, 15, 234-281.

A MAP INTERFACE FOR EXPLORING MULTIVARIATE PALEOCLIMATE DATA

David DiBiase, Catherine Reeves, Alan M. MacEachren, John B. Krygier, Martin von Wyss, James L. Sloan II* and Mark C. Detweiler**

Department of Geography
The Pennsylvania State University
302 Walker Building
University Park PA 16802
Email: dibiase@essc.psu.edu

*Earth System Science Center
Penn State

**Department of Psychology
Penn State

ABSTRACT

We begin with an abbreviated review of recent approaches to the display of three or more geographically-referenced data variables from the literatures of statistics, computer graphics and cartography. In comparison we describe a method we have developed for exploring relationships among multivariate paleoclimate data produced by a global circulation model at Penn State's Earth System Science Center. Key features of the interface include 1) display of multivariate data in two-dimensional, planimetrically-correct map formats; 2) optional display of a single map on which up to three variables are superimposed or four maps among which up to four variables are juxtaposed; and 3) the ability to quickly revise data variable selections, to focus on subsets of the data and to experiment with different combinations of point, line and area symbols.

CONTEXT

This paper discusses methods for displaying three or more geographically-referenced data variables. In particular we are concerned with methods appropriate for exploratory data analysis in Earth system science. Earth system science is an integrative, multidisciplinary approach to understanding "the entire Earth system on a global scale by describing how its component parts and interactions have evolved, how they function, and how they may be expected to continue to evolve on all time scales" (Earth System Sciences Committee 1988). Numerical modeling is an important method of formalizing knowledge of the behavior of the Earth system and for predicting natural and human-induced changes in its behavior. The volume of quantitative data produced by model simulations and earth observing satellites imparts a high value on effective graphical techniques for identifying potentially meaningful patterns and anomalies. Because the Earth system comprises many interrelated phenomena, the ability to display multiple data variables in a format that fosters comparison is particularly desirable.

In the following we will first discuss several recent approaches to the display of three or more variables from the literatures of statistics, computer graphics and cartography. Then we describe an interface we have developed for displaying up to four paleoclimate data sets produced by a global climate model at Penn State's Earth System Science Center. Key features of the interface include 1) display of multivariate data in two-dimensional, planimetrically-correct map formats; 2) op-

tional display of a single map on which up to three variables are superimposed or four maps among which up to four variables are juxtaposed; and 3) the ability to quickly revise data variable selections, to focus on subsets of the data and to experiment with different combinations of point, line and area symbols. In this approach we have attempted to incorporate the three functional characteristics that differentiate cartographic visualization from cartographic communication: the interface is tailored to the needs of a specific set of users, its intended use is more to foster discovery rather than to present conclusions, and it is an interactive environment that encourages users to experiment with different combinations of data and graphic symbols (MacEachren in press). We conclude with a brief discussion of the problem of evaluating the effectiveness of this and other multivariate exploratory methods.

EXPLORATORY MULTIVARIATE DATA ANALYSIS

Most display media for computer-based analysis available to geoscientists and analysts in other disciplines provide two-dimensional images. Primary among these are raster CRTs and paper or film imaging devices. Three-dimensional illusions can be created by employing psychological or physiological depth cues (Kraak 1988). Though 3-D offers higher information-carrying capacity than 2-D, adding a dimension to the depiction also introduces additional cognitive and computational costs. The relative advantages of 2-D versus 3-D depend on the dimensionality of the phenomena to be visualized (and their surrogates, the data variables), and whether relations among different variables are assumed to be orthogonal or parallel. Also pertinent are the hardware and software available to the analyst, the cognitive tasks to be performed and his or her aesthetic preferences.

Two-dimensional approaches One class of 2-D approaches exploits "glyphs"—compound point symbols such as faces, trees, castles, and the "icons" used in the Exvis system developed by Grinstein and colleagues at the University of Lowell (Smith and others 1991). An Exvis display consists of a dense 2-D matrix of strings of short line segments. The orientation of each line segment relative to those connected to it represents a quantity for one of several variables at a given location. Remarkably differentiable patterns can emerge as textural differences in properly "focused" Exvis displays. The system provides user control for the size of each icon, line segment length, line segment orientation limits, and displacement of icons to reduce artifactual patterns.

For analysts willing to suspend the orthogonality of longitude and latitude, relations among many geographic variables can be viewed at once using parallel coordinate diagrams (Bolorforoush and Wegman 1988). In this method each variable is assigned a parallel linear scale. Corresponding cases on each scale are connected by straight lines. Positive and negative correlations, clusters, and modal cases among many variables can be revealed in patterns of parallelism, convergence, and central tendency among the connecting lines. When the number of cases is large, the density of overlapping connecting lines can be interpolated and displayed as isolines (Miller and Wegman 1991). Coordinate scales must often be normalized or transformed in other ways to produce meaningful patterns.

Cartographers and geographers also have demonstrated display methods for multivariate quantitative data on 2-D maps. Most efforts by cartographers to date have been directed to maps intended for presentation rather than exploratory use. Exceptions include Bertin's (1981) "trichromatic procedure" for superimposing three point symbol maps of quantitative data colored with the three printers' primaries (cyan, magenta and yellow) whose combination results in an informative range of hues and lightnesses. Dorling (1992) produced a time series of population cartograms of England superimposed with arrows symbolizing the mixture of alliances to three political parties (by hue and saturation) and the magnitude and direction of voting swings (by length and orientation) in more than 600 constituencies in Parliamentary elections.

Three-dimensional approaches Three-dimensional representations are invaluable for portraying volumetric phenomena. Describing the animated stereoscopic display terminals developed for use with the McIDAS weather display system, Hibbard (1986) remarks that "[t]hey create an illusion of a moving three-dimensional model of the atmosphere so vivid that you feel you can reach into the display and touch it." Interactive 3-D displays are also desirable for investigating forms of relationships among non-spatial variables in 3-D scatterplots (Becker and others 1988). For map-based displays, however, 3-D does not necessarily simplify the problem of effectively superimposing three or more data distributions.

In a review article on meteorological visualization, Papathomas and colleagues (1988) note that two approaches to multivariate 3-D visualization have been attempted: "One is to portray each variable by a different attribute [graphic variable] ... another is to assign different transparency indices to the various surfaces that represent the variables." The latter approach has proven limited for multivariate display. As Hibbard (1986) observed, "[w]hen viewing a transparent surface and an underlying opaque surface simultaneously, the eye is quite good at separating the shade variations of each surface and deducing their shapes. However, this breaks down when there are several transparent layers." The former method has been exploited successfully in a revealing simulation of the cyclogenesis of a severe storm by Wilhelmson and his colleagues at the National Center for Supercomputing Applications at the University of Illinois (Wilhelmson and others 1990). Tracers, streamers, buoyant balls and translucent shading are artfully combined to reveal storm dynamics. Display of 3-D imagery can be costly, however: the seven and one-half minute simulation required the equivalent of one person's full-time effort for more than a year and 200 hours rendering time on a Silicon Graphics Iris workstation.

Realism vs. abstraction Papathomas and colleagues (1988) remark that "[b]oth of the above methods result in images that are highly 'unrealistic,' illustrating that there may be instances in scientific computing in which the visualization technique may have to transcend 'realism.'" More recently, several authors have argued that increasing abstraction rather than realism may often be an effective visualization strategy (Muehrcke 1990, MacEachren and Ganter 1990). Scientists seem skeptical about this argument, however. We hypothesize that desire for realistic 3-D imagery in earth science visualization is often motivated by the aesthetic appeal of

ray-traced 3-D imagery and the intuitive metaphor of the terrain surface as a vehicle for expressing statistical surfaces. Robertson (1990) has elaborated this metaphor as a "natural scene paradigm." In our view, the uncritical acceptance of this paradigm will sometimes lead to both ineffective data representations and unnecessarily expensive visualization hardware and software.

Multiple views Exploratory analyses involve a variety of tasks. Carr and colleagues (1986) observe that "[i]n general, we do not know how to produce plots that are optimal for any particular perceptual task." Since a data display may be called upon to serve several tasks, a search for a single optimal display method is likely to be futile. Some (for example, Monmonier 1991) have argued that single displays created to communicate data relationships to the public may even be unethical.

Windowing standards such as X-Windows now make it easy to produce multiple data displays for exploratory analysis. Two strategies are constant formats (for example, a sequence of maps of the same extent and scale showing changes in time series data) and complementary formats (such as a scatterplot linked to a map).

Tufte (1990) refers to ordered series of information graphics displayed in a constant format as "small multiples," and asserts that "[f]or a wide range of problems in data presentation, small multiples are the best solution." Bertin (1981), who generalizes the variety of visualization tasks as two types of questions (What is at a given place? and Where is a given characteristic?), declares that "it is not possible, with complex information, to represent several characteristics in a comprehensive way on a single map while simultaneously providing a visual answer to our two types of question." Only comparison of several juxtaposed univariate maps can answer both types, he argues.

Monmonier has adapted several multiple view, complementary format display techniques for geographic analysis. In 1989 he described the concept of "geographic brushing" as an extension of scatterplot brushing (Becker and others 1988). Geographic brushing of trivariate distributions involves linked display of scatterplot matrices or 3-D scatterplots (whose axes represent thematic attributes) and a map. Extending the work of Gabriel (1971), Monmonier (1991) also describes the "geographic biplot" as "a two-dimensional graphic on which a set of place points represents enumeration areas or sample locations and a set of measurement points represents different indicators or different years for a single measure." The biplot's axes are usually derived from principal components analysis. Attributes are plotted by eigenvector coefficients, places by component scores. "Relative proximity on the biplot reflects relative similarity among places and variables." The biplot is most effective when two principal components account for a large proportion of the variation in a multivariate data set. When three axes are required to explain a satisfactory fraction of variation, 3-D biplots can be constructed (Gabriel and others 1986).

Single superimposed views Superimpositions (single displays on which symbols representing three or more variables are overlaid) are preferred for some exploratory analyses. Synoptic climatologists, for example, rely on multivariate

maps to characterize interactions of atmospheric phenomena that give rise to severe weather events. One example is a series of seven-variable superimpositions published by Lanicci and Warner (1991). Their maps employ a strategy for multivariate superimposition suggested by Bertin (1981), in which the discriminability of overlaid distributions is maximized by expressing different data variables with symbols of different dimensions (point, line and area). This strategy has also been employed by Crawfis and Allison (1991) for their "scientific visualization synthesizer."

Dynamic graphics A recurring theme in the literatures we have consulted is the advantage of dynamic graphical methods over their static counterparts. For most analysts, the term "dynamic graphics" connotes interaction: the "direct manipulation of graphical elements on a computer screen and virtually instantaneous change of elements" (Becker and others 1987). Both fast hardware and an interface that promotes efficient performance of intended tasks are required for fruitful interaction. Highly interactive systems allow users to easily select among lists of data variables to be displayed, different display formats and symbolization options. In object-oriented displays, individual elements or groups of elements may also be selected. Several operations on selected elements may be provided. For superimposed map-based displays, an analyst may want to be informed of the values of the multiple variables at a selected location. Analysts using linked, complementary format displays (as in brushing) need selections on one display to be automatically highlighted on the other.

Another important interaction task is rotation. User-controlled coordinate transformations that appear to rotate viewed objects are usually required to discover structure in multivariate distributions displayed as surfaces, point clouds and translucent volumes. Some systems allow users to specify real-time coordinate transformations by which analysts can guide a 2-D viewing plane through n-dimensional data spaces. Young and Rheingans (1990), for instance, demonstrated a real-time "guided tour" through a six-dimensional data space derived from principal components analysis of rates of seven types of crime for the United States. Cubes representing the states were "rocked" back and forth between two 3-D biplots formed by the first three and last three principal component coefficients, respectively. Clusters in the six-dimensional space were revealed by similar patterns of movement among the cubes as the biplots were rocked.

The abbreviated review which space permits us here cannot do justice to the variety and ingenuity of methods devised for the display of three or more geographically-referenced data variables. We hope that it provides at least a backdrop for the interface development project we describe next.

CASE STUDY: EXPLORATORY MULTIVARIATE ANALYSIS OF PALEOCLIMATE DATA

The project we describe here is being carried out in the Deasy GeoGraphics Laboratory, a cartographic design studio affiliated with the Department of Geography that serves the Department, the College of Earth and Mineral Sciences and The Pennsylvania State University. One of the leading clients of the laboratory over the past five years has been Penn State's Earth System Science Center (ESSC), a consortium

of 23 geoscientists, meteorologists and geographers who study the global water cycle, biogeochemical cycles, earth's history and human impacts on the earth system. ESSC is one of 29 interdisciplinary teams participating in NASA's Earth Observing System (EOS) research initiative.

Recently, in response to emerging demand for animated and interactive multimedia presentations, Deasy GeoGraphics (like similar publication-oriented labs at other universities) has adopted a new emphasis on software development. When asked how the lab might apply its cartographic expertise in a development project that would benefit ESSC analysts, ESSC's Director replied that he had long wished to be able to simultaneously display multiple spatial "fields" on a single map base.

Typically, ESSC researchers rely on juxtaposed comparison of laser-printed univariate isoline maps to assess the spatial correspondence of model-produced atmospheric and oceanographic distributions. ESSC staff describe how the Director had not long ago performed a synoptic analysis of Cretaceous precipitation, upwelling and winter storms for an award-winning paper (Barron and others 1988) by overlaying paper maps on his office window.

Overlay analysis is a generic GIS problem. More specifically, however, the problem we have addressed is how to effectively display multiple spatial distributions for ESSC researchers' exploratory analyses. This problem is not susceptible to generic solutions.

Paleoclimate data Our interface displays data produced by numerical models such as the National Center for Atmospheric Research Community Climate Model (CCM). The CCM is a three-dimensional climate model whose resolution is 7.5° in longitude by approximately 4.5° latitude, yielding a regular grid of 1,920 predictions. At ESSC, the Genesis version of the CCM runs on a Cray Y-MP2 supercomputer. A typical application of the model is to simulate three to five climatic phenomena for nine to twelve atmospheric levels for the Cretaceous (100 million years before present). The product of the simulation is a digital "history volume" which consumes about two gigabytes of disk space or tape in binary form. Analysts subsequently use a separate software module called the CCM Processor to produce "history save tapes" in which the model-produced data may be aggregated to monthly, seasonal or yearly averages. One kind of history save tape is a "horizontal slice tape," an IEEE-format binary file containing aggregate univariate data predicted for a plane that intersects the atmosphere parallel to Earth's surface. The data are quantitative, interval/ratio scaled, gridded, 2-D representations of continuous atmospheric phenomena.

We have christened our map interface "SLCViewer" to denote its role in the ESSC software toolbox as an interactive, mouse-driven utility for viewing multiple horizontal slice tapes. We developed SLCViewer using the Interactive Data Language (IDL) on a Sun Sparc2 workstation under Unix. The project is currently in the "proof of concept" stage. Procedures developed in IDL may be implemented later in other visualization or GIS applications if warranted.

The SLCViewer interface(s) SLCViewer actually offers analysts a choice of two interfaces: a single map or four juxtaposed maps. After launching IDL, then entering "slcview" to compile a main call program, the SLCViewer base widget

appears. The base widget provides two buttons named "one" and "four" by which users select an interface. The base widget also provides pop-up menus for selecting data variables, geologic time periods, and for specifying isoline levels and colors. Current selections are displayed textually. An "exclusion" button calls up another widget for focusing on subsets of selected data. We'll discuss focusing a little later.

The analyst moves his or her mouse pointer from the base widget to the map window to symbolize the selected data. SLCViewer can display data with point, line and area symbols on a cylindrical equidistant map projection (which despite its shortcomings is used commonly in the global change community). Figure 1 shows three variables superimposed in the single map window. Symbol designation controls are identical for both the one map and four map windows.

Focusing on subsets of the data Our work has been influenced by two ESSC scientists who have met with us on several occasions to observe our progress and offer suggestions. Early on they outlined the sequence of exploratory tasks they expected to perform with the interface. The sequence involves initial inspection of univariate maps of complete distributions, followed by focused observation of selected subsets of distributions (a process they referred to as "exclusion"), then superimposition of several complete or subsetted variables. SLCViewer supports the focusing process by providing an exclusions widget by which users can restrict displays to values greater than or less than a specified threshold value. Excluded

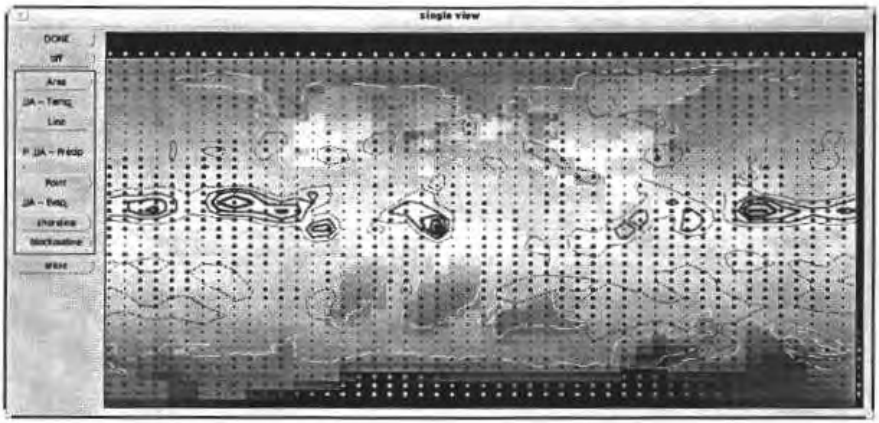


Figure 1 Single-map window of the SLCViewer interface in which three paleoclimate variables are superimposed (38 percent of original size). Average June, July and August temperature data for the Cretaceous (100 million years before present) have been mapped onto a range of 100 gray levels and are displayed as a matrix of gray areas that correspond to the spatial resolution of the model. Lighter areas represent warmer temperatures. Average precipitation data for the same period have been interpolated and are depicted by weighted isolines that increase in width for higher values. Average evaporation is represented by four categories of point symbols classified by nested means. Light and dark points (red and blue in the original) represent data above and below the mean. Large and small points designate the outer and inner two categories.

values in data sets symbolized with area (gray scale) symbols are represented with dark blue grid cells. Isolines can be suppressed via the exclusions widget or simply by specifying the desired isoline levels in the lines widget. The exclusion widget does not withhold data from interpolation but merely suppresses display of specified isolines. Exclusions do not cause any point symbols to disappear, but a specified value (such as zero) replaces the mean of the data set in the classification of symbol sizes and colors.

Two-dimensional displays SLCViewer displays multivariate paleoclimate data in 2-D, planimetrically-correct map formats. Although the CCM is a 3-D model of the atmosphere, the horizontal slice tapes produced by the CCM Processor are 2-D matrices representing predicted values for discrete horizontal planes in the atmosphere. Following Tufte's (1983) maxim that the dimensionality of a graphic representation should not exceed the dimensionality of the data, 2-D data ought to be represented by 2-D images. His assertion can be extended for display of multiple horizontal slice tapes. Since each horizontal slice is parallel to Earth's surface, all horizontal slices are parallel to one another. When the slices chosen for combined display represent phenomena at the same level of the atmosphere (as they usually do), they are coplanar. When these conditions are met, superimposed 2-D map displays are faithful representations of multivariate paleoclimate data.

Given the good sense of Tufte's principle, and partial support for it in empirical research (see Barfield and Robless 1989 and Kraak 1988), we have chosen to portray multiple 2-D data fields on 2-D, planimetric map bases. A secondary practical benefit of the 2-D approach is that substantially less computing power is required for interactive performance, meaning that analysts need not have access to very high performance workstations to use SLCViewer effectively.

Multiple views SLCViewer provides optional display of a single map on which up to three variables are superimposed or four maps among which up to four variables are juxtaposed. In an interactive exploratory environment the choice between single and multiple displays need not be exclusive. SLCViewer allows analysts to have it both ways. The single map interface facilitates superimposition of three variables (or more, if the analyst is willing to deal with multiple isoline or point displays). The four-map interface provides small multiples. Although we are keen on the notion of multiple complementary formats, we have restricted our current efforts to developing map-based strategies.

Interactivity Finally, SLCViewer is a dynamic environment in which analysts may select data variables to be displayed and express each variable with a distinctive set of point, line or area symbols. The ability to quickly revise data variable combinations, to focus on subsets of the data and to try different symbolization strategies could increase the efficiency of exploratory analyses of model data at ESSC. By easing the exploratory process we hope that SLCViewer may even foster new insights. The ESSC scientists who have consulted with us in SLCViewer's development expect that the interface will be useful also as a teaching tool in geoscience classes.

Symbol dimensions and graphic variables The trivariate map strategy we adopted for the single map interface involves contrasting the dimensionality of

symbols (point, line and area) and enhancing each symbol type with the graphic variables that are most potent for expressing quantitative data (size for lines and points, lightness for area symbols). This approach contrasts with the iconographic approach used for Exvis, another 2-D method in which all variables are represented as compound line symbols that vary in size, orientation and color. Though Exvis displays excel in revealing overall patterns among multiple variables, interpretation of patterns in the distribution of individual variables seems difficult. We have not yet begun to formally evaluate the effectiveness of our strategy, however.

EVALUATING EFFECTIVENESS

Two crucial tests of the effectiveness of an exploratory technique are whether it is used at all, and if so, whether its use profits the analyst. If comparable methods exist, these can be compared in controlled studies. Such studies are valuable since visualization is expensive both in terms of time and capital outlays. Benefits ought to be documented. The initial development of SLCViewer continues even as this article takes shape—the interface has not yet been used by ESSC scientists for actual research. Our plan is to promote SLCViewer within ESSC and, assuming it is adopted, closely monitor the results.

CONCLUSION

With long experience in the visual display of quantitative information, cartographers have much to contribute to the practice of scientific visualization. We only occasionally encountered references to cartographic research in the literatures we consulted, however. In an earlier publication, some of us called on cartographers to engage their experience and creativity in applied visualization projects and to observe carefully what scientists find useful (DiBiase and others 1992). In this project we have aimed to practice what we preached. Our efforts have been guided by the needs and expertise of scientists among whom visualization is a routine but inefficient activity. Our contribution is a prototype exploratory multivariate data analysis tool for geographically-referenced data that maintains the planimetric integrity and computational efficiency of 2-D maps, offers analysts a choice of single or multiple views, and provides user control over data variable and graphic symbol selections. We cannot yet gauge the value of this contribution. We'll see.

ACKNOWLEDGMENTS

Thanks to Bill Peterson, Jim Leous and Jeff Beuchler of the ESSC Computing Facility who provided us the workstation, advice on IDL and access to the slice tape data. We appreciate also the interest of ESSC researchers Karen Bice and Peter Fawcett, and Eric Barron's unflagging support.

REFERENCES

Barfield, W. and R. Robless (1989). "The effects of two- and three-dimensional graphics on the problem-solving performance of experienced and novice decision makers", *Behaviour and Information Technology* 8(5), pp. 369-385.

Barron, E.J., L. Cirbus-Sloan, E. Kruijs, W.H. Peterson, R.A. Shinn and J.L. Sloan II (1988). "Implications of Cretaceous climate for patterns of sedimentation", Annual Convention of the American Association of Petroleum Geologists, Houston, TX, March 1988.

Becker, R.A., W.S. Cleveland and A.R. Wilks (1987) "Dynamic graphics for data analysis", *Statistical Science* 2, pp. 355-395, reprinted in W.S. Cleveland and M.E. McGill (1988) *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth; pp. 1-50.

Becker, R.A., W.S. Cleveland and G. Weil (1988) "The use of brushing and rotation for data analysis", in W.S. Cleveland and M.E. McGill (eds) *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth; pp. 247-275.
Bertin, J. (1981) *Graphics and Graphic Information Processing*. New York: deGruyter.
Bolorforoush, M. and E.J. Wegman (1988) "On some graphical representations of multivariate data", in E.J. Wegman, D.T. Gantz and J.J. Miller (eds), Computing *Science and Statisitics, Proceedings of the 20th Symposium on the Interface* pp. 121-126.
Carr, D.B, W.L. Nicholson, R.J. Littlefield and D.L. Hall (1986) "Interactive color display methods for multivariate data", in Wegman, E.J. and DePriest (eds), *Statistical Image Processing and Graphics*. New York: Marcel Dekker.
Crawfis, R.A. and M.J. Allison (1991) "A scientific visualization synthesizer", in G.M. Mielson and L. Rosenblum (eds), *Proceedings Visualization '91*, Los Alamitos, CA: IEEE Computer Society Press, pp. 262-267.
DiBiase, D., A.M. MacEachren, J.B. Krygier and C. Reeves (1992) "Animation and the role of map design in scientific visualization", *Cartography and Geographic Information Systems* 19(4), pp. 201-214.
Dorling, D. (1992). "Stretching space and splicing time: from cartographic animation to interactive visualization", *Cartography and Geographic Information Systems* 19(4), pp. 215-227.
Earth System Sciences Committee, NASA Advisory Council (1988) *Earth System Science: A Closer View*, Washington DC: The National Aeronautics and Space Administration.
Gabriel, K.R. (1971) "The biplot graphic display of matrices with application to principal-component analysis", *Biometrika* 58, pp. 453-467.
Gabriel, K.R., A. Basu, C.L.Odoroff and T.M. Therneau (1986) "Interactive color graphic display of data by 3-D biplots", in T.J. Boardman (ed), *Computer Science and Statistics—Proceedings of the 18th Symposium of the Interface*. Washington, DC: American Statistical Association, pp. 175-178.
Grinstein, G. and S. Smith (1990) "The perceptualization of scientific data", in E.J. Farrell (ed), *Extracting Meaning from Complex Data: Processing, Display, Interaction*, Proceedings SPIE (International Society for Optical Engineering) 1259, pp. 164-175.
Hibbard, W.L. (1986) "Computer-generated imagery for 4-D meteorological data", *Bulletin of the American Meteorological Society* 67(11), pp. 1362-1369.
Kraak, M.J. (1988) *Computer-assisted Cartographical Three-dimensional Imaging Techniques*, Delft, the Netherlands: Delft University Press.
Lanicci, J.M and T.T. Warner (1991) "A synoptic climatology of the elevated mixed-layer inversion over the southern plains in spring. Part II: The life cycle of the lid", *Weather and Forecasting* 6(2), pp. 198-213.
McCleary, G.F. (1983) "An effective graphic 'vocabulary'", *Computer Graphics and Applications* March/April, pp. 46-53.
MacEachren, A.M. and J.H. Ganter (1990) "A pattern identification approach to cartographic visualization", *Cartographica* 27(2), pp. 64-81.
MacEachren, A.M. (in press) "Cartography-cubed", in A.M. MacEachren and D.R.F. Taylor (eds) *Visualization in Modern Cartography*, Oxford, England: Pergamon Press.
Miller, J.J. and E.J. Wegman (1991). "Construction of line densities for parallel coordinate plots", in A. Buja and P.A. Tukey (eds), *Computing and Graphics in Statistics*. New York: Springer-Verlag
Monmonier, M (1989) "Geographic brushing: enhancing exploratory analysis of the scatterplot matrix", *Geographical Analysis* 21(1), pp. 81-84.
Monmonier, M (1990) "Strategies for the visualization of geographic time-series data", *Cartographica* 27(1), pp. 30-45.
Monmonier, M. (1991) "Ethics and map design: confronting the one-map solution", *Cartographic Perspectives* 10, pp. 3-7.
Muehrcke, P.C. (1990) "Cartography and geographic information systems", *Cartography and Geographic Information Systems* 17(1), pp. 7-15.
Papathomas, T.V, J.A. Schiavone and B. Julesz (1988) "Applications of computer graphics to the visualization of meteorological data", *Computer Graphics* 22(4), pp. 327-334.
Robertson, P.K. (1990) "A methodology for scientific data visualization: choosing representations based on a natural scene paradigm", *Proceedings of the First IEEE Conference on Visualization, Visualization '90*, pp. 114-123.
Smith, S., G. Grinstein and R. Pickett (1991) "Global geometric, sound and color controls for iconographic displays of scientific data", in E.J. Farrell (ed), *Extracting Meaning from Complex Data: Processing, Display, Interaction II*, Proceedings SPIE (International Society for Optical Engineering) 1459, pp. 192-206.
Tufte, E.R. (1983) *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.
Tufte, E.R. (1990) *Envisioning Information*. Cheshire, CT: Graphics Press.
Wilhelmson, R.B., B.F. Jewett, C. Shaw, L.J. Wicker, M. Arrott, C.B. Bushell, M. Bauk, J. Thingvold and J.B. Yost (1990) "A study of a numerically modeled storm", *The International Journal of Supercomputing Applications* 4(2), pp. 20-36.
Young, F.W. and P. Rheingans (1990) "Dynamic statistical graphics techniques for exploring the structure of multivariate data", in E.J. Farrell (ed), *Extracting Meaning from Complex Data: Processing, Display, Interaction*, Proceedings SPIE (International Society for Optical Engineering) 1259, pp. 164-175.

INTELLIGENT ANALYSIS OF URBAN SPACE PATTERNS: GRAPHICAL INTERFACES TO PRECEDENT DATABASES FOR URBAN DESIGN

Alan Penn
University College London
The Bartlett School of Architecture and Planning,
Gower Street, London WC1E 6BT, England
email: alanp@bartlett.ucl.ac.uk

ABSTRACT

Where urban models of the 1960's and 70's were criticised for lack of data, Geographic Information Systems are now being criticised for a lack of analysis. The problem is that there is a fundamental lack of theory of how complex urban systems work, and of how they relate to the spatial and physical structure of cities. New techniques of Configurational Analysis of Space (CAS) based on Hillier's 'space syntax' theories (Hillier & Hanson, 1984) describe cities as patterns of interconnected spaces. By describing and quantifying pattern properties the techniques allow researchers to investigate the way urban function and spatial design relate. One of the main findings of this research is that the pattern of the street grid is a primary determinant of patterns of both pedestrian and vehicular movement. It appears that, given their head, urban land use patterns evolve to take advantage of movement and the opportunities this creates for transaction. The theoretical understanding of urban systems which is being developed using the new analytic techniques is now being applied in design. By building together interactive analysis of space patterns with databases of existing cities and urban areas in which we understand the principles governing function, knowledge of precedent is brought to bear on design.

USEFUL THEORY AND THE CREATION OF PHENOMENA

It is a characteristic of most fields of study that deal with human and social systems that practice runs well ahead of theory. There is perhaps no better illustration of this than in urban planning and design where theory, following the disasters constructed in its name during the hey-day of modernism, is widely held to be a discredited force. Practice must move on with or without the help of theory, and the 'applicability gap' between research and application in the social sciences is often cited as the reason why more often than not practice makes do without theory. It was not always so. Amongst the strongest remnants of the scientific optimism of modernism are the urban models developed during the late fifties and sixties. Transport modelling at least remains a strong force (some would say too strong) in design practice today. But modelling seems not to have borne out its promise as the generalised tool for urban planning. Why is this? Michael Batty in his review of some thirty years of urban modelling (Batty 1989) suggests that it is because the social and political context that gave rise to modelling in the first place has disappeared. Models were, he suggests, a product of planners working towards a 'collective or public interest'. In a period in which individualism and self interest are in the ascendancy the very intellectual underpinning of modelling as tool for rational redistribution has disappeared. In its absence, he argues, modelling has all but died as a policy tool.

In a sense this is ironic. One of the main problems to dog modelling in its early years was a lack of data. Today data sources are expanding more rapidly than ever before. Private sector data providers have taken over as governments have reduced their efforts, and the storage and retrieval of spatial data using Geographic Information Systems has never been easier. But GIS is itself being criticised for failing to synthesise data into useful knowledge (Openshaw, Cross & Charlton, 1990). I believe the failure of urban models to move with the times and answer current questions and the criticisms of GIS for lack of application follow from the same root cause: that no one has defined what 'useful knowledge' is in the context of spatial systems. In this paper I discuss just what form 'useful knowledge' might take in terms of the questions designers and urban

masterplanners ask. I suggest that for design the two most important forms of knowledge are knowledge of precedent and knowledge of context. The utility of knowledge derives from its ability to allow one to act intentionally. I will argue that for knowledge to be useful it must be theoretical in the sense that it must allow one to predict. Scientific theories and scientific knowledge have just this property. I believe that Batty's calls for a stronger theoretical base for urban modelling and Openshaw's for a stronger analytic base for GIS amount to the same thing. They are both asking for urban phenomena to be put on a firm scientific basis in which knowledge will allow reasonable predictions to be made, and so will allow actions to be intentional. The suggestion is that it is only when theory becomes useful in this sense that it will begin to influence practice.

On the face of it these calls for a stronger scientific basis for urban analysis seem contradictory. From the beginning urban models *looked* very scientific. They are dressed in complex equations and based on concepts like gravity and equilibria with a respectable history in physics and chemistry. But those that criticised these models for their lack of data missed their real weakness. It was not that they lacked data, it was that they failed for the most part to generate their theories from the data. Geographic Information Systems appear to suffer from the reverse problem. They too look very scientific in that they gather data in huge quantities - something that we know scientists do a lot of. But they are now being criticised for a lack of analytic capability. I believe the main problem is not that they lack analysis, on the contrary they are highly flexible analytic tools. It is that the analysis they provide is if anything of too general application. Given data in a GIS the user is left pretty much to their own devices to decide how best to add value to that data. In this sense it seems that the form of the analysis they provide does not follow closely enough from they kinds of problem they are being asked to help solve. Perhaps the contradiction, and so our sense of dissatisfaction with both urban modelling and increasingly with GIS, is based on the fact that each is bedded in a particular paradigmatic view of science. Where urban modelling fulfilled an essentially Popperian hypothesis testing programme - the hypothesis or model comes first and the data second as a check or calibration of the model - GIS provides for an inductive approach led by the provision and manipulation of raw data in the hope that theory will emerge.

Ian Hacking in an influential recent contribution to the philosophy of science (Hacking 1983) suggests a resolution of these two opposed views of the scientific process. He proposes that one of the activities of science that is all too often overlooked is 'the creation of phenomena' which then become the subject matter for theory. He is talking mainly of the work of experimentalists. The people who like to tinker and produce phenomena - regular occurrences that seem consistent but often baffle explanation - and which provide the driving force behind new theory. He describes this activity in a way that sets it apart from mere data gathering. Creating phenomena requires the active manipulation of data, finding new ways of representing things so that regularities are revealed (it is important that phenomena are visible), measuring and calculating so that you can quantify what you see.

I believe that what urban theory lacks at the moment are the regular phenomena about which to theorise. Some phenomena like the regular motion of the planets, as Hacking notes, are open to observation. Others have to be worked at and actively created. Useful urban theory will only come about if we first create some phenomena about which to theorise. In a subject area which is social and evolutionary, and in which the physical objects we deal with cover areas of hundreds of square kilometres, the computer is likely to be a useful tool in creating phenomena. It is particularly good at calculating and transforming representations, and these are important aspects of the creation of phenomena.

SPATIAL CONFIGURATION AND URBAN PHENOMENA

The concentration of econometric models on the 'process' aspects of urban function has led them to overlook, or at best simplify, their representation of the spatial and physical form of cities to the point at which it ceases to be an input to the process. For designers, though, the spatial and physical city is the major concern - what they need to know is 'what difference does it make if I design this one way or another?' Conversely, where GIS is

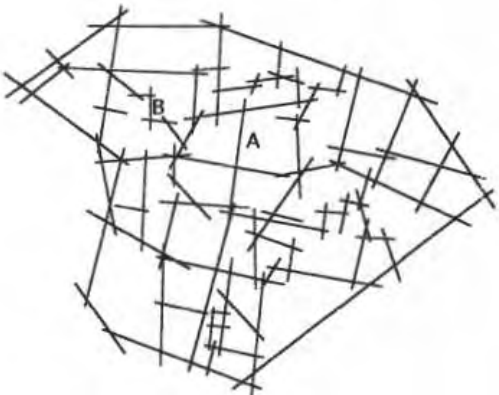
concerned, representations of social and economic processes are almost entirely lacking. The description of nearly all aspects of urban form is possible, but without a basis in knowing which aspects are important in urban processes it is all just so much data. What seems to be lacking is an urban theory relating the physical city and the social processes that take place within it.

In the late 70's Bill Hillier began to develop just such a theory. Central to his efforts was the notion that an understanding of urban systems requires a description not just of individual elements of the city - this building or that space - but a description of whole systems of spaces and buildings considered as patterns. Hillier and our group at UCL have developed representational and analytic techniques which combine the geographer's concern for located data with the modeller's concerns for processes. They are based on a simplified description of space, neither as the geographer's 2-D homogeneous plane which is accessible 'as the crow flies', nor as the modeller's arbitrary zones and links, but in terms that obey the real physical constraints that built form places on visibility and movement since you can neither see nor move through solid objects, only through open space. Figure 1. shows something of what I mean. 1a. is the map of a French market town; 1b. shows the pattern of open space in the town coloured in black. This pattern of space is the part of the town that people use and move through. It is the object we are interested in studying. We are looking for ways of representing this irregular pattern that carry some social 'potential'; 1c. picks out the main 'convex' elements of space in the town, and plots the fields of view - all space that can be seen from those spaces. As can be seen, the open squares are all visually interlinked by more or less linear pieces of space; 1d. then passes the longest lines of sight and access through the open space pattern - it essentially simplifies the plan into a series of linear spaces of the sort that approximate shortest and simplest routes between different parts of the town.

Figure 1a)

1b)

1c)



1d)

Marked on the line map 1d are two locations 'A' and 'B'. Both are close to each other and relatively central in the town, but in terms of their locations relative to the whole town considered as a configuration they are radically different. What do I mean by a 'configuration'? A simple transformation of the representation illustrates the phenomenon. Figure 2a represents line 'A' as a dot at the bottom of a graph, the five lines that intersect with line 'A' are represented as dots one level above, those that intersect with each of the five appear at level 2, and so on until all the lines in the town have been represented. In other words, the graph represents the minimum number of changes of direction that need to be made to get from line 'A' to anywhere else in the town, since each move from one line to another involves a change of direction. Figure 2b does the same thing from the point of view of line 'B'. The two graphs are very different. From line 'A' the rest of the town is relatively shallow. That is, it is accessible by making fewer changes of direction than from line 'B'. On the basis of this we can say that considered in terms of their relationship to the whole configuration the two lines are different. The property of depth is 'global' in that it relates a line to its whole context.

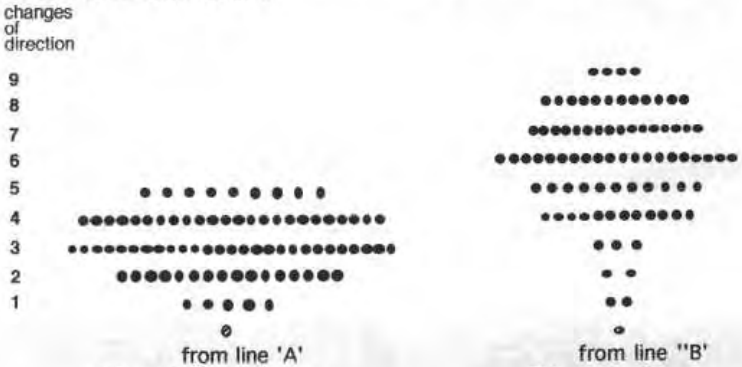


Figure 2a) 2b)

The difference between 'A' and 'B'is clearly measurable: all we have to do is measure the mean 'depth' of the graph from the point of view of each line in turn and we have a global, relational measure of how each piece of linear space in the town relates to all others. This exercise has been carried out and the depths represented back on the line map in Figure 3 as a greyscale from black for shallowest through to light grey for deepest.

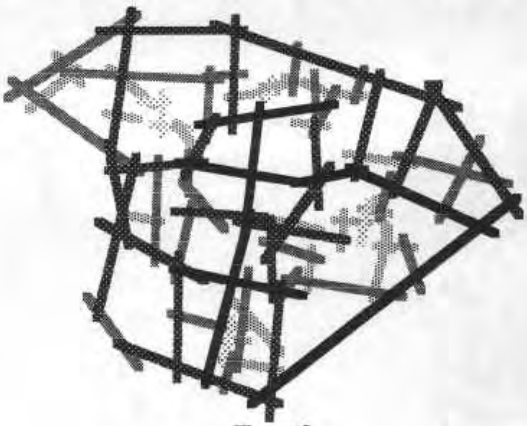


Figure 3

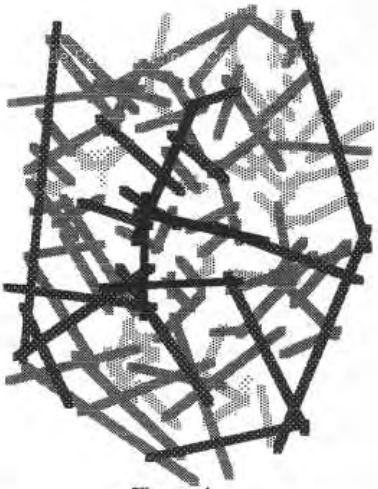
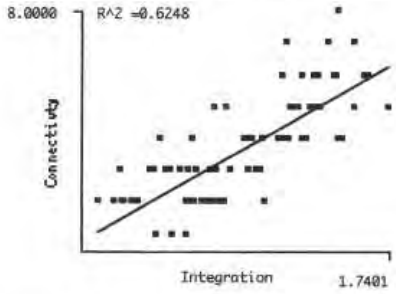


Figure 4

We call the measure of shallowness in the graph (relativised to take account of the number of lines in the system) 'spatial integration' since it quantifies the degree to which a linear space or street integrates or is segregated from its urban context. The shape distribution of the back 'integrated' streets forms a 'deformed wheel' with part of the rim, a number of spokes leading from edge to centre in several directions and a discontinuity at the hub. The more segregated areas lie in the interstices of this integration structure. The pattern is quite characteristic of a large class of settlements from all parts of the world. Figure 4, for example, is the integration core of a hutted settlement from southwestern Africa in which a 'deformed wheel' of shallow space allows easy and controlled access for visitors from outside to the centre town and back out again.

The similarities go deeper than just the distribution of integration. In both settlements there is a strong relationship between local properties of space, such as the number of other lines each line is connected to, and the global property of integration. We call the correlation between local and global spatial properties the 'intelligibility' of a system since it gives an index of the degree to which it is possible to predict where you are in terms of the whole configuration (everything that you cannot see) on the basis of what you can see locally. Figure 5a and b show the intelligibility scattergrams for the two settlements shown in Figures 3 and 4 respectively.



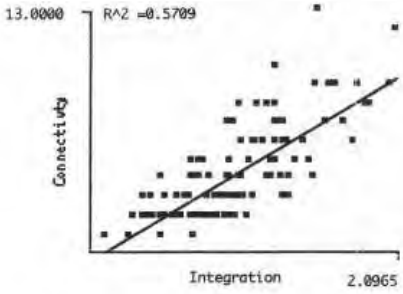


Figure 5a) 5b)

Again we find that intelligibility is a general property of a large class of 'organic' settlements of this size. This is not a trivial finding. It is simple to design settlements in

which the relationship between local and global is broken, and significantly we find this has been done in many of the problematic recent public housing schemes in the UK which are often referred to by residents as 'maze like'. Although we have no direct psychological evidence, is tempting to suggest that the human mind may act as a 'correlation detector' in activities such as spatial searches and wayfinding. Where no correlations can be found confusion may result.

NATURAL MOVEMENT AND THE MOVEMENT ECONOMY

One of the other most common accusations made against our modern housing schemes is that they lack 'vitality' and at worst turn into 'urban deserts' devoid of people even in the middle of the day. It is a longstanding question whether the design of the schemes is in any way at fault in this, or whether the blame rests with the way that the schemes are managed, with the socioeconomic status of the residents and so on. The multivariate nature of this sort of socio-spatial question has made it seem almost impossible to resolve one way or the other. However, the new techniques for describing and quantifying the geometric and topological form of urban space allow us to approach some of the simpler questions at the heart of this issue.

Since we can now describe and quantify radically different spatial designs on the same basis we can begin to 'control' the design variable in studies of other aspects of urban function. It is possible to detect effects of spatial design on patterns of pedestrian movement simply by observing pedestrian flow rates at a number of locations in the streetgrid and then using simple bivariate statistics to look for the relationship between configurational measures of those locations and flows. A large number of studies has now established that spatial integration is consistently the strongest predictor of pedestrian flow rates (see Hillier et al, 1993, for a comprehensive review of the evidence). Spatially integrated lines carry greater pedestrian flows than more segregated ones. The effects are strong and consistent. Figure 6, for example shows the line map for an area of north London in which we have made detailed observations of pedestrian movement patterns. Each street segment has been observed for a total of over 50 minutes at different times of day and on different days of the week. The all day mean hourly pedestrian flow is noted on each segment. Figure 7 shows the scattergram of a measure of 'local integration' in a much larger model of the area against the log of pedestrian flow rates. The 'local integration' value is measured in the same way as global integration but restricting the number of lines considered to those within three changes of direction. In this case the model is much larger than that shown, extending approximately two kilometres away from the observation area in all directions. The correlation is remarkably strong at $r=.872$, $p<.0001$.

The key discovery here is that the correlation is between pedestrian flows and a purely spatial measure of the pattern of the streetgrid. No account has been taken of the location of attractors or generators of movement in constructing the measure of spatial integration. It seems that movement patterns result naturally from the way the spatial configuration of the streetgrid organises simplest routes (involving fewest changes of direction) to and from all locations in an area (Penn & Dalton, 1992). Of course this runs counter to the premises of urban modelling which hold that the key facts in urban systems are the distributions of activities and land uses that 'generate' or 'attract' flows between different geographic locations. Our results leave us in little doubt that the primary fact is the pattern of space, and that if there is a relationship between land uses and pedestrian flows (which there certainly is - you find more people on streets with shops than on streets without) this is most likely to be due to retailers choosing their shop sites with care in order to take advantage of the opportunities for passing trade provided by the natural movement pattern resulting from the grid.

We find support for this hypothesis when we look at samples of shopping and non-shopping streets (Hillier et al 1993). Consistently we find that in areas that include shopping streets there is an exponential increase in pedestrian flows with integration (hence the linearisation of the pedestrian rates in Figure 7 using a logarithmic scale). In non-shopping areas, however, the correlation is predominantly linear. A possible explanation for this would invoke a mechanism in which shops locate themselves in numbers that are in

proportion to the level of passing trade generated by the pattern of the grid. The shops then attract a number of additional trips in proportion to the attractiveness of the shop. We might then expect areas including shopping to exhibit a multiplier on the basic pattern of natural movement that would be consistent with an exponential growth in pedestrian flows.

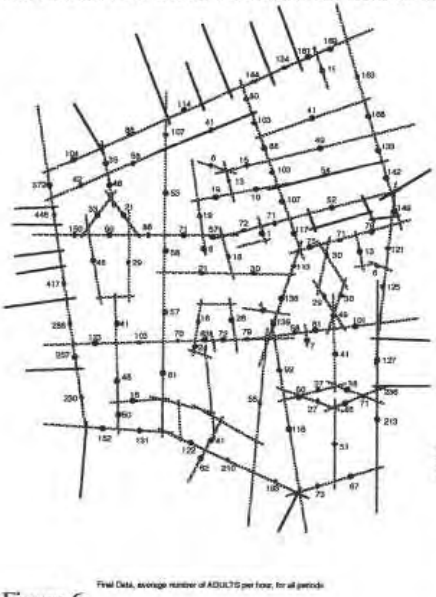


Figure 6

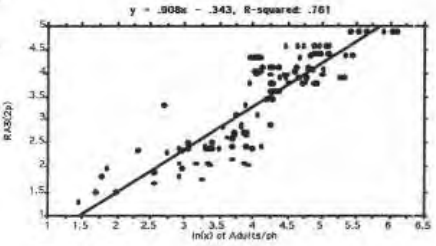


Figure 7

There is a marked shift in emphasis from existing urban theory suggested by these findings. Where current theory takes activities and land uses as the prime movers in urban systems the new results suggest that the spatial configuration of the city exerts systemic effects on both land use and flows. A new economics of the city would move from considering movement as a cost, to movement as a resource that brings potential transactions past land parcels. Trips are as important in terms of what they take you past as they are in terms of where they take you from or to. This moves the scope of urban theory from a subjective individual view in which origins and destinations are the important facts, to an objective social theory in which individual trips are primarily of interest in that they add together to create a probabilistic field of potential encounter and transaction which other members of society can take advantage of in deciding their actions. This suggests an economics of the city based on movement as its principal resource and spatial configuration as its main implement (Hillier & Penn 1989).

THE PROBABILISTIC INTERFACE AND SOCIAL PATHOLOGY

One of the most pertinent criticisms of urban models focuses on their apparent lack of a conception of human or social life in cities. They are criticised for taking a mechanistic view of urban systems in which the missing elements are the people. Conversely, where the geography of urban pathology has been studied by looking at clusters of social malaise (most recently by Colman, 1985) the main criticism has been that of mistaking correlation for causality, and that the mechanisms through which design was supposed to be related to social malaise were either disregarded or unsubstantiated (Hillier 1986). GIS apparently faces a dilemma as it tries to incorporate more extensive modelling capabilities. It risks either concentrating on mechanism to the exclusion of all else, or disregarding mechanism to the point at which its results mislead. I believe one resolution of this problem lies in results of recent studies of space use and abuse in housing areas.

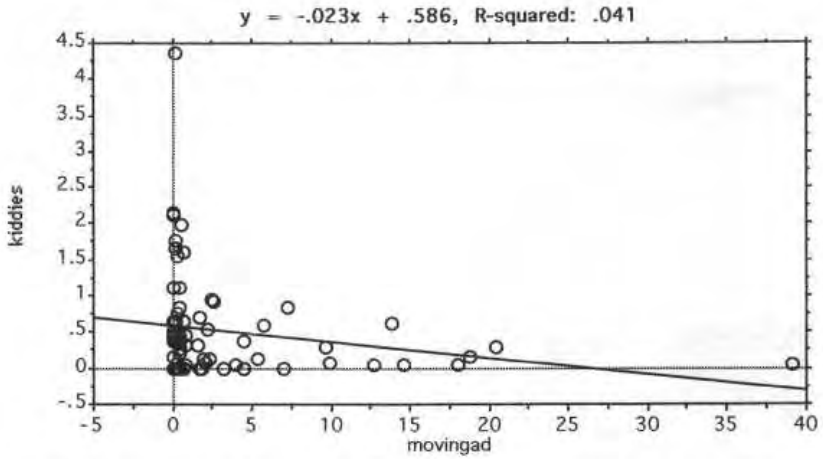


Figure 8. The correlation between adults and children in a public housing scheme

There is remarkable consistency in the way that post war public housing in the UK has sought to segregate its public open space from the surrounding street fabric. This is demonstrated by substantially reduced integration values in for housing estate spaces compared to traditional street areas in configurational analyses. As might be expected, presence of pedestrians also drops substantially in estate interiors, to the point at which they are sometimes referred to as 'deserts' or being in a state of 'perpetual night'. It seems that spatial segregation serves to isolate the estates from all through movement to the point at which you can be alone in space for most of the time. However, where we observe space use patterns by different categories of people simultaneously we find still more suggestive results. Patterns of space use by children and teenagers of school age differ radically from those of adults. Children gather in groups, often not moving much but using space to 'hang out' in locally strategic locations which are cut off from the outside world in the estate interior.

These locations tend to remove them from informal overseeing by adults as they move into and out of the estate, and if we look at the correlation between adult and child presence in estate interiors we find a characteristic 'L-shaped' distribution shown in Figure 8. Where there are greater numbers of adults there are low numbers of children, and where there are larger numbers of children there are lower numbers of adults. In normal urban streets there is a much stronger correlation between adults and children suggesting that an informal interface is maintained. These findings are now being added to by more recent studies of crime locations which suggest that the strategic locations in estate interiors which are emptied of normal adult levels of movement by being segregated from through movement to and from surrounding areas become the favoured locations for specific types of crime and abuse.

It seems quite possible that the configuration of urban space through its effects on patterns of movement may construct informal probabilistic interfaces between different categories of people. The interface between shop owners and buyers makes transaction possible, that between adults and children may turn socialisation and control into natural and unforced processes. Alternatively, where space structures lead to a polarisation of space use by different social categories, we suspect that distrust, stigmatisation and crime result. It seems possible given this view of the relation between social processes and spatial configuration that the theories which gave rise to zoning of 'communities' in their own 'territories' served to create the social pathologi es they intended to control. If this is so it is little wonder that 'theory' has gained such a poor reputation amongst practitioners and the public alike.

APPLYING KNOWLEDGE IN DESIGN

Making theory useful in design depends above all on answering questions posed in the form of physical designs. Designers are good at generating physical and spatial 'form'. They find it more difficult to judge the likely outcomes of that form in terms of all the interacting criteria they are expected to handle in a functioning urban area. In most fields they employ engineers to advise on outcomes in specific domains - structures, air movement and so on. However, when it comes to social outcomes they have no one to refer to. This is where configurational analysis techniques come into their own. Since they analyse spatial patterns directly and in some detail they are able to analyse designs on the drawing board as well as existing urban areas. Essentially, the analysis speaks in the same terms that the designer uses to design. To the extent that research findings are general across all the areas that have been studied previously, advice can be given on the expected outcomes of physical design in human and social terms. The principle of natural movement following spatial integration is a case in point. Following an analysis of a design, predictions can be made of likely patterns of pedestrian movement. These can then be used to ensure adequate levels of space use to make areas safe as well as to make sure that land uses are located to take advantage of the pattern of natural movement. To the extent that findings are specific to particular locations or areas, advice can only be given if studies have been made of the particular context for a scheme. Two types of knowledge are therefore useful in design: knowledge of precedent and knowledge of context.

The whole programme of research at UCL is geared to this. By carrying out consultancy projects for design teams engaged in live projects research questions are kept in line with those that designers need to answer. The spin off is an ever increasing database of studied areas each with its own set of specific design problems. By carrying out funded research the methodologies and software are developed to keep abreast of the questions emerging from practice. A typical consultancy project requires the construction of a large area context model for the development site in question. Figure 9 shows the context model of Shanghai recently constructed for Sir Richard Rogers' competition winning scheme for the new financial capital for China. Next the model is amended to include the new design and the analysis is rerun. On the basis of the analysis advice is given in the light of knowledge of precedent from other cities and of the specific context of Shanghai. The design is then amended and the analysis rerun in a process akin to hypothesis testing.

Figure 9. The integration map for Shanghai

In this particular case the aim of the design was to synthesise a whole range of issues from energy usage to social contact and microclimate in order to create a sustainable low

energy city for the next millennium. The programme for the design was therefore to obtain the maximum output for a given input in the pursuit of social, economic and environmental sustainability. The arena within which the new techniques are being developed is thus quite different from the 'rational redistribution' goals that drove the first urban models. Instead, the aims are to use an understanding of the processes which drive urban systems to develop new urban forms that function in a natural way requiring the minimum intervention. I believe that this sets the new CAS techniques apart from traditional urban modelling approaches which found their roots in the interventionist planning ideology of the 60's.

The new graphic computer technologies available today make these aims achievable. The creation of visual phenomena is central to the development of theory as well as to its application in practice. Graphic devices that internalise and make visible knowledge of precedent, such as the depth graph, the integration pattern or the intelligibility scattergram, are vital tools in communicating to designers and the lay public. People like to see the results of design changes and to have somthing to work towards. Engineers have been able to provide this for some time in the domains of air movement and structures, but it is only now that we are beginning to provide the same kinds of graphic and interactive capability in the more social domains of design.

REFERENCES

Batty M.,1989, Urban Modelling and Planning: Reflections, retrodictions and prescriptions, from Macmillan B. (ed) Remodelling Geography, Basil Blackwell, Oxford, 147-169.

Colman A., 1985, Utopia on trial: vision and reality in planned housing, Hilary Shipman, London.

Hacking I., 1983, Representing and Intervening: Introductory topics in the philosophy of natural science, CUP, Cambridge.

Hillier B. & Hanson J., 1984, The Social Logic of Space, CUP, Cambridge, England.

Hillier B., 1986, City of Alice's Dreams, Architects Journal, London, 9th July 1986, 39-41.

Hillier B. & Penn A., 1989, Dense Civilisations: or the shape of cities in the 21st century, Applied Energy, Elsevier, London.

Hillier B., Penn A., Hanson J., Grajewski T., Xu J., 1993, Natural Movement: or configuration and attraction in urban pedestrian movement, Planning and Design: Environment and Planning B, Pion, London.

Openshaw S., Cross A. & Charlton M., 1990, Using a supercomputer to improve GIS, proceedings of Mapping Awareness.

Penn A. & Dalton N, 1992, The Architecture of Society: stochastic simulations of urban pedestrian movement, Surrey Conferences on Sociological Methods: Simulating Societies, University of Surrey, Guilford.

The Geographer's Desktop: A Direct-Manipulation User Interface for Map Overlay*

(Extended Abstract)

Max J. Egenhofer and James R. Richards
National Center for Geographic Information and Analysis
and
Department of Surveying Engineering
University of Maine
Boardman Hall
Orono, ME 04469-5711, U.S.A.
{max, richards}@grouse.umesve.maine.edu

Abstract

Many spatially aware professionals use the manual process of map overlay to perform tasks that could be done with a GIS. For instance, they could be using GIS technology for work in environmental sciences and design fields, however, they are often not doing so because they lack the computer expertise necessary to run a GIS. The user interface of current GISs has been frequently cited as a major impediment for a broader use of GISs. Popularity and success of metaphor in other areas of human-computer interaction suggests that visual, direct manipulation user interfaces are especially attractive and easy-to-learn for non-computer experts. Many GISs use map overlay as a command-line based interaction paradigm. An interface to GIS that is a visualization of the map-overlay metaphor would enable experts in the spatially aware environmental sciences to more easily use GIS as a regular tool. To overcome this shortcoming, a new direct manipulation user interface based on the map-overlay metaphor has been designed and prototyped. It is well embedded within the successful Macintosh desktop and employs the particular characteristics of metaphor, direct manipulation, and iconic visualization. We create a *geographer's desktop* by replacing the familiar notions of files and folders with the concepts of *map layers* and a *viewing platform* on which layers can be stacked. A visualization of this user interface is presented. Particular attention is given to the way users can change the symbology of layers placed on the viewing platform.

Introduction

Many geographic information systems (GISs) attempt to imitate the manual process of laying transparent map layers over one another on a light table and analyzing the resulting configurations. While this concept of *map overlay*, familiar to many geo-scientists, has been used as a design principle for the underlying architecture of GISs, it has not yet been visually manifested at the user interface. To overcome this shortcoming, a new direct manipulation user interface for overlay-based GISs based on the map-overlay metaphor has been designed and prototyped. It is embedded within the successful Macintosh desktop metaphor and employs the particular characteristics of metaphor, direct manipulation, and iconic visualization. We create a geographer's desktop by replacing the familiar notions of files and folders with the concepts of *map layers* and a *viewing platform* onto which layers can be stacked. This goes beyond the concepts present in the user interfaces of popular

* This work was partially supported through the NCGIA by NSF grant No. SES-8810917. Additionally, Max Egenhofer's work is partially supported by NSF grant No. IRI-9309230, a grant from Intergraph Corporation, and a University of Maine Summer Faculty Research Grant.

GISs such as ARC/INFO and MAP II, as we build the user interface on a coherent metaphor *and* employ direct-manipulation techniques to perform actions, rather than circumscribing the actions by constructing sentences.

This paper presents the concepts and a visualization of this user interface in which each layer is represented by a single icon and the operations of modifying their content and graphical representation are achieved by double clicking on different parts of the icon. This differs from a previously discussed visualization (Egenhofer and Richards 1993), in which the information of a layer was split into two separate icons for its content and its graphical presentation to address individually what to retrieve from a geographic database and how to display the information (Frank 1992).

The remainder of this paper is structured as follows: The next section briefly reviews prior efforts in designing user interfaces for map-overlay based GISs and introduces as an alternative the concepts of the Geographer's Desktop. Subsequently, a particular visualization of the Geographer's Desktop is discussed, for which a sequence of interface snapshots are given. The conclusions point towards future research.

The Geographer's Desktop

The theoretical background of this user-interface design is based on a number of previous investigations. First, there is Tomlin's MAP algebra (Tomlin 1990), an informal collection of operations to be performed on layers of maps. A number of formalizations of the map-overlay concept have been given, e.g., as C++ code (Chan and White 1987) or as algebraic specifications (Dorenbeck and Egenhofer 1991). The idea of map overlay is complemented by investigations into GIS query languages and the recognition that there are two distinct issues to be addressed by a spatial query language (Egenhofer 1991): (1) the specification of what to retrieve (the *database retrieval*) and (2) the specification of how to present the query result (the *display specification*).

The second area influencing this work comprises several innovative studies of how to present map overlay at the user interface, that go beyond the typing of fairly complex commands (ESRI 1990) and the translation of commands into selections from pull-down menus (Pazner *et al.* 1989). For example, a graphical version of MAP II (Kirby and Pazner 1990) provides an approach to map overlay in which a user composes an "iconic sentence" through a sequence of icons that are dragged over the screen. At the same interaction level, GeoLineus (Lanter and Essinger 1991) allows a user to put together icons for layers, displaying on the screen the sequence of operations as specified in a command language—including the ability to track the composition of an overlay operation. These approaches are improvements over the typed version of a map algebra. They reduce the complexity of composing an overlay as the user need not remember a particular command, but can rather identify and select it from a set of given operations. On the other hand, both approaches retain the inherent structure of a map-algebra operation, which any user has to follow. As such, users are constructing sentences in the map algebra, but not necessarily performing a particular problem-solving task.

As an alternative to the command-line based languages for map-overlay, we pursue visual, direct-manipulation languages based on the map-overlay metaphor. In such a language, users perform actions much like the professional experts do in their familiar environment. They also receive immediate feedback about the status of operations being performed. This approach is superior to languages that describe operations verbally, because it reduces the "gulf of execution," e.g., the extra effort required of users to successfully complete a task) at the user interface (Norman 1988).

The *geographer's desktop* is a direct-manipulation user interface for map overlay (Frank 1992; Egenhofer and Richards 1993). The principal components of the geographer's desktop are (1) *geographic data* that a user can select and combine and (2) *presentation parameters* that apply to the data selected. The selection of data corresponds to a database query with which a user specifies which subset of the data is available to display. Geographic data can be combined with a variety of analytical tools, though initially the geographer's desktop will be restricted to the fairly simple combination of *graphical overlay*. A number of different analysis methods are available for geographic data, e.g., graphical (i.e., map-like) display of spatial information, tabular presentation (much like the common presentation of business data), or statistical charts.

The geographer's desktop manifests the map-overlay metaphor by adding a *viewing platform*, which is linked with a viewing window. The viewing platform represents the light table in the source domain. Users place layers onto the platform in order to see the contents of the layers as rendered by the appropriate visualization parameters. or remove them from the platform to erase them from the current view. The viewing platform has several functionalities: (1) It enables a direct-manipulation implementation of map-overlay and acts much like the trash can on the Macintosh user interface, which allows users to delete (and recover) files by dragging them into (and out of) the trash can. Likewise, selecting and unselecting geographic data becomes a direct-manipulation operation by dragging map layers onto and off the viewing platform, respectively. (2) The viewing platform allows a user to distinguish between a house-keeping operation on a direct-manipulation user interface and an operation to add spatial data to the set of viewable data, or to remove spatial data from the currently visible data set.

Layer Visualization of the Geographer's Desktop

Different direct-manipulation implementations of visualizing and interacting with the relevant objects are possible for the Geographer's Desktop. In one visualization, the database query and symbolization components have been treated as separate objects on the user interface surface (Frank 1992; Egenhofer and Richards 1993). Here a different approach is pursued as the database query and symbolization components are visually linked. This section introduces the two principal objects that comprise the *layer visualization* and details the operations that are typically performed on them. The objects are (1) layers and (2) viewing platforms. This is followed by a discussion of the visualization and interaction of the major operations. The user interface visualization presented is not merely a computerized version of map-overlay, but rather it exploits the primary concepts involved, while allowing for metaphor extensions to utilize advantages available in a computer environment.

Layers

The concept of *layers* is borrowed from the map-overlay metaphor, where physical maps exist as map layers. Generally, physical maps almost always have a legend and an area for map drawing. The legend shows the map reader what the various map symbols are, and the map drawing contains the graphic rendering of the map as defined by that symbolization. In a digital environment, the legend can correspond to the symbolization component and the drawing can correspond to the database query component as rendered by the symbolization rules. These concepts are brought together to form the *layer object* at the user interface (Figure 1).

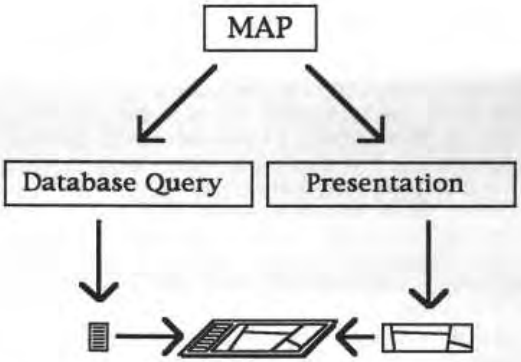
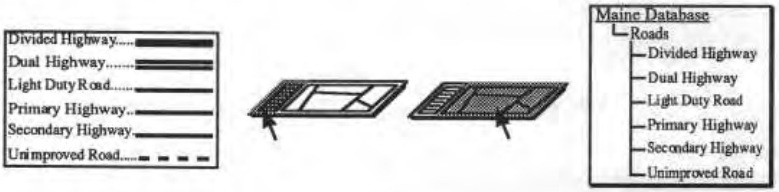


Figure 1: Visual integration of the components into a layer icon.

At the surface, the layer object is an icon that consists of two halves, with which the user can interact separately. Double clicking on either the legend for symbolization (Figure 2a) or the map for database query (Figure 2b) engages the user in a dialog for interacting with those parameters.



(a) (b)

Figure 2: Double clicking on the left part of the layer icon (a) allows the user to manipulate the symbology, while double clicking on the right part of the icon (b) allows the user to modify the database selection.

Viewing Platform

When manipulating objects on the geographer's desktop, users must have a way of distinguishing between actions intended for "house cleaning" (i.e., moving things around), and actions meant to initiate operations upon the objects. The viewing platform is the object that enables users to differentiate these two fundamentally different kinds of actions (Figure 3).

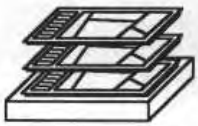


Figure 3: A viewing platform with three layers stacked on top.

Platforms are associated with a set of viewing windows in which the multiple views of the data may be displayed concurrently. New viewing platforms can be created on demand so

that users may simultaneously compare several different overlays. Each platform can have multiple "hot linked" windows that correspond to the multiple views of the data. In a hot linked window system, different views of the data are displayed concurrently in different windows. Direct manipulation queries can be performed in one window with the results being shown in all.

Adding and Removing Layers

Initially, layers are located anywhere on the geographer's desktop (Figure 4).

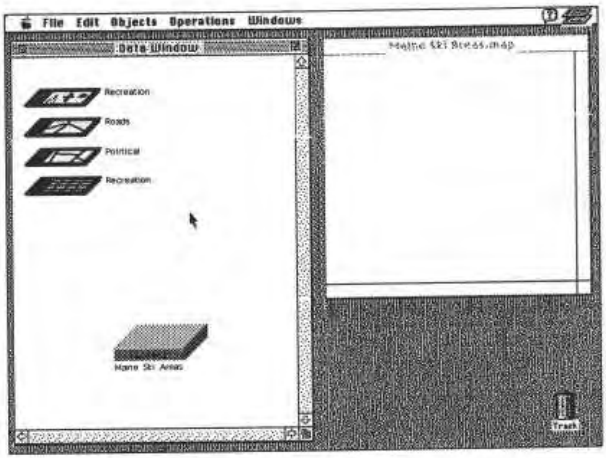


Figure 4: A snapshot of the layer visualization of the geographer's desktop.

A user selects the layer(s) of interest with a direct manipulation device and combines them by dragging them across the desktop onto the viewing platform in order to view them. Placing a layer on top of a viewing platform will initiate the rendering of its database query component as specified by its symbolization component. The result will then appear in the platform's appropriate view window. The layer may specify different kinds of renderings such as graphic, statistical, or tabular. The snapshot in Figure 5, for instance, shows the geographer's desktop after three layers with a graphic rendering and one with a tabular rendering had been put onto the viewing platform. The graphical and the tabular results are displayed in corresponding view windows.

Layers can be dropped onto the viewing platform by releasing the mouse in the zone above it and the platform will attract the layers, like a magnetic force, and stack them neatly on top of itself. During this process, users receive feedback about the status of their operation: the platform informs when the release of the mouse will result in the desired actions (i.e., before the mouse button is actually released) by highlighting when the mouse moving the selected layers is in the zone above the platform. The induced activity of attracting the layers once they are within the platform's area of influence is very similar to that of the trash can on the Macintosh desktop, which highlights when a file is dragged over it and absorbs that file when the mouse button is released. Reversely, when removing a layer, the platform's feedback informs the user that the selected layer or group of layers has been removed successfully or not. Removed layers remain at the location on the desktop where the user places them.

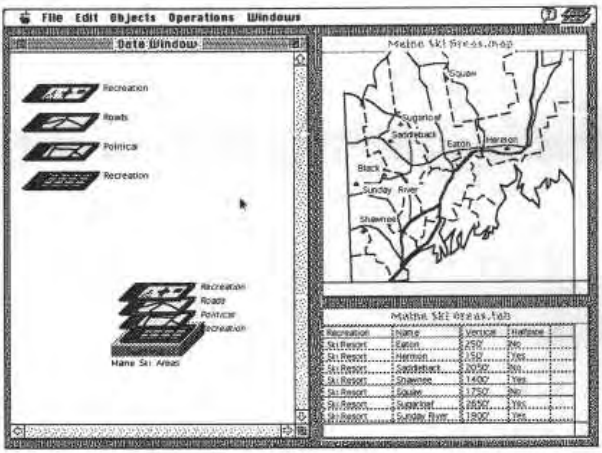


Figure 5: The geographer's desktop after four layers had been moved onto the viewing platform.

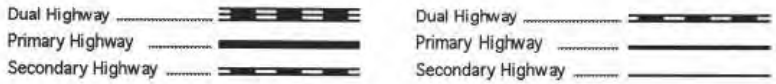
Modifying the Symbology

In addition to being the location where layers are placed to be viewable, the viewing platform includes the functionality of manipulating the symbolization components. Such symbolizations may form complex cartographic renderings according to a predefined map style such as "USGS 1:24,000."

The concepts that the implementation of this functionality is based on are similar to the feature known as *style sheets* in Microsoft Word (Microsoft 1991). Style sheets allow users to save specifications of how text should be formatted. There are interactive methods that users employ in order to create and use various style sheets. When a user first launches the application, there is a default font, size, and tab setup that is called the Normal style. A user who makes changes to Normal-styled text can then afterwards do two different things with a selection. First, the user can select some text whose style has been altered, and then choose "Normal" from the styles pop-up menu. At this point the user would see a dialog box asking whether to "Reapply the style to the selection," or "Redefine the style based on the selection." Choosing the first option would cause the selected text to reassume the default style "Normal," while choosing the second option would cause the definition of the style "Normal" to assume the form of the selected text's format. This, in turn, causes all text in the document that is styled "Normal" to assume the newly defined characteristics. Second, the user can select some reformatted text and choose **Style...** from the **Format** menu. This action produces a dialog box which allows the user to name this new style and add it to the list of styles available in the styles pop-up menu. After dismissing the dialog with an OK, the user could then select other pieces of text and reformat them by choosing the desired style from the styles menu.

The layers of this user interface visualization have similar features that can be explained easily via a comparison with MS Word. There are several default styles available to the user, which could include "USGS 1:24,000," "Rand McNally Road Map," user defined styles, and others. With the creation of a new layer, the user must choose both a default style and a database query. For example, a user may define a layer "roads" which uses the

USGS 1:24,000 roads symbolization (Figure 6a), with a database query that includes only Dual Highways, Primary Highways, and Secondary Highways. Alterations of the symbolization are possible via interaction with the legend portion of the layer. For example, if the user decided to make road symbolization smaller, it might look like Figure 6b.



(a) (b)

Figure 6: (a) USGS 1:24,000 roads symbolization and (b) a user-defined roads symbolization.

At this point the user could perform any of the same operations that are possible with MS Word styles. For instance, after choosing "USGS 1:24,000 Roads" from a styles menu, the user would be asked whether to "Reapply the style to the selection," or "Redefine the style based on the selection." Choosing the first option would revert the style back to its original form, while choosing the second would make all instances within the given document that used the USGS 1:24,000 Roads style assume the new symbolization. Second, the user could choose a **Styles...** command from a menu, allowing the naming, saving, and new availability of the style. It might be called "Minor USGS Roads," or something to that effect. After performing this task, the new style would then be available to any other layer with a roads database query within the current document.

In addition to styles within individual layers, the viewing platform can have an associated *Style Collection*, which is indicated by a name on the front of the platform. Default collections would consist only of the individual style components that make up "USGS 1:24,000," "Road Map," user defined styles, and others. For example, a USGS 1:24,000 collection would include "USGS 1:24,000 Hydrology," "USGS 1:24,000 Roads," etc.

Conflicts arise and must be resolved when a user drags a layer with a different style onto a platform. If, in the above example, the user had changed the roads symbolization of a layer and then dragged that layer onto a platform with a default "USGS 1:24,000" Collection, she or he would be presented with a dialog asking for clarification of the action. The first option would be to "Reapply 'USGS 1:24,000 Roads' to the user defined symbolization," and the second option would be to "Begin a new Collection." If the first option were chosen, the symbolization in the given layer would revert back to the default "USGS 1:24,000 Roads" style, and if the second option were chosen, the platform would begin accepting layers with different kinds of styles and putting them in a new Collection. Subsequently choosing the **Collection...** command from a menu would allow the user to name, save, and make available the new Collection. The user might give this new collection a descriptive name such as "USGS with Minor Roads."

The platform, with its many different Collections, allows the user the opportunity to quickly compare different renderings of the same layers. Changes in symbolization are easily accomplished by choosing a Collection from the pop-up menu on the front of the viewing platform—e.g., from USGS 1:24,000 to City Council (Figure 7). In addition, the viewing platform provides the user with feedback about the selection of the symbolization. With a pop-up menu on the front the user can manipulate style collections. The menu displays the current style collection of a given platform. When users change style collections by adding layers with different visualizations than the current collection, a new collection is started that is based on an extension of the first collection. This is visualized by adding a "++" to the end of the current style collection on the pop-up menu. Once the user

has established a number of style collections, changing between them is as simple as selecting the desired style from the pop-up menu.

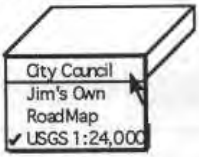


Figure 7: Changing the symbology by choosing a Collection from the viewing platform.

Conclusions

GIS user interface design has been established as a major topic in recent years (Mark et al. 1992). This paper contributes to this area with work in the visualization and interaction process of user interface design. This emphasis was possible because of the framework within which the work was done, where a formalization of data cubes and templates was already in existence (Frank, 1991). The geographer's desktop is a framework for visualizing a user interface to geographic information, in which map overlay was presented as the metaphor for interaction in the design.

The interaction concepts have been studied in a mockup created with the animation application MacroMind Director™ (MacroMind, 1991). The feedback we received from viewers was generally very positive. They stressed that the direct-manipulation visualization of the map-overlay metaphor is very intuitive. They also found it very appealing and powerful to have the flexibility of manipulating individually the database selection and symbolization components.

A number of questions are still open, some of which are currently under investigation. For instance, how can computational overlay be included within the framework of map overlay? The user interface as designed and visualized thus far addresses only graphic overlay; however, one of the strengths of GIS is its capability to perform such computational overlays as intersections and buffering operations. One viable solution would be to combine the map-overlay metaphor with a visualization such as grade school *addition*, which is a source metaphor with which virtually everyone is familiar (Figure 8).

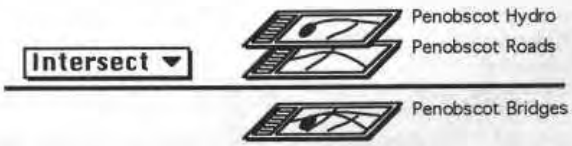


Figure 8: The simple *addition metaphor* applied to more complex geographic computational overlay.

When can certain metaphors be combined? Besides map overlay, there are other metaphors in use for such applications as panning and zooming in geographic space. Investigations into how these and other metaphors can coexist seamlessly in a GIS, are necessary in order to promote more metaphor-based interactions and implementations in commercial GISs.

Acknowledgments

Andrew Frank was instrumental in the design of the Geographer's Desktop with the "data cubes and maps" user interface. Discussions with David Mark, Gary Volta, Doug Flewelling, and Todd Rowell provided further useful input. Kathleen Hornsby helped with the preparation of the manuscript. Thanks also to countless visitors for their feedback during prototype demonstrations of earlier versions of the Geographer's Desktop.

References

K. Chan and D. White (1987) Map Algebra: An Object-Oriented Implementation. *International Geographic Information Systems (IGIS) Symposium: The Research Agenda*, Arlington, VA, pp. 127-150.

C. Dorenbeck and M. Egenhofer (1991) Algebraic Optimization of Combined Overlay Operations. in: D. Mark and D. White (Eds.), *Autocarto 10*, Baltimore, MD, pp. 296-312.

M. Egenhofer (1991) Extending SQL for Cartographic Display. *Cartography and Geographic Information Systems* 18(4): 230-245.

M. Egenhofer and J. Richards (1993) Exploratory Access to Geographic Data Based on the Map-Overlay Metaphor. *Journal of Visual Languages and Computing* 4(2): 105-125.

ESRI (1990) *Understanding GIS—The ARC/INFO Method*. ESRI, Redlands, CA.

A. Frank (1992) Beyond Query Languages for Geographic Databases: Data Cubes and Maps. in: G. Gambosi, M. Scholl, and H.-W. Six (Eds.), *Geographic Database Management Systems. Esprit Basic Research Series* pp. 5-17, Springer-Verlag, New York, NY.

K.C. Kirby and M. Pazner (1990) Graphic Map Algebra. in: K. Brassel and H. Kishimoto (Eds.), *Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp. 413-422.

D. Lanter and R. Essinger (1991) *User-Centered Graphical User Interface Design for GIS*. Technical Report 91-6, National Center for Geographic Information and Analysis, University of California at Santa Barbara, Santa Barbara, CA.

MacroMind (1991) *MacroMind Director™ 3.0 User's Manual* MacroMind Inc., San Francisco, CA.

D. Mark (1992) *Research Initiative 13 Report on the Specialist Meeting: User Interfaces for Geographic Information Systems*, Technical Report 92-3, National Center for Geographic Information and Analysis, University of California at Santa Barbara, Santa Barbara, CA.

Microsoft (1991) *User's Guide Microsoft WORD, Word Processing Program for the Macintosh, Version 5.0*. Microsoft Corporation, Redmond, WA.

D. Norman (1988) *The Design of Everyday Things*. Doubleday, New York, NY.

M. Pazner, K.C. Kirby, and N. Thies (1989) *MAP II: Map Processor—A Geographic Information System for the Macintosh*. John Wiley & Sons, New York, NY.

C.D. Tomlin (1990) *Geographic Information Systems and Cartographic Modeling*. Prentice-Hall, Englewood Cliffs, NJ.

EMPIRICAL COMPARISON OF TWO LINE ENHANCEMENT METHODS

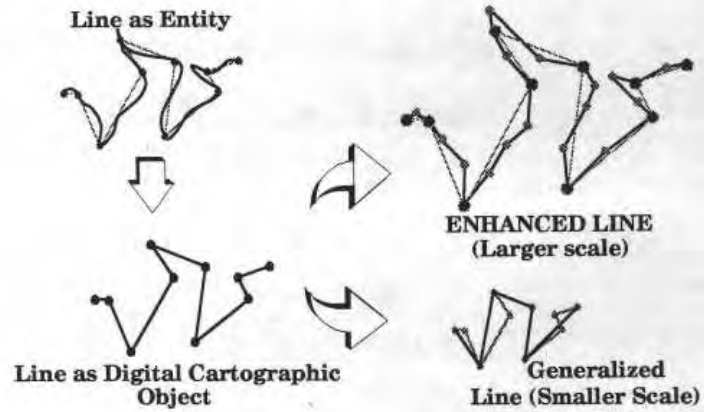
Keith C. Clarke, Richard Cippoletti, and Greg Olsen
Department of Geology and Geography
Hunter College--CUNY
695 Park Avenue
New York, NY 10021, USA
e-mail: kclarke@everest.hunter.cuny.edu

ABSTRACT

Considerable recent cartographic research has focussed upon the theory, methods, and cartometry associated with the generalization of line features on maps. The inverse of the generalization transformation, enhancement, has received comparatively little attention with a few notable exceptions which use fractal methods. This study implements Dutton's enhancement method and uses the equivalent theory but an alternative method (Fourier Analysis) to present a parallel, but more analytically flexible technique. While the Fourier method requires an initial equidistant resampling of the points along a string, the reduced number of control parameters, and the shape-related metadata evident in the Fourier transform make the method an attractive alternative to the Dutton technique, particularly when further resampling is used. Test data sets from Dutton's work and for the Long Island, New York coastline were used to demonstrate and empirically compare the two methods.

INTRODUCTION

Cartography has been called a "discipline in reduction," in that the mapping sciences deal exclusively with the case where display takes place at scales smaller than those at which the data are captured, stored, or analyzed. Illustration of this concept follows from the consideration of a line feature on a map.



As an "entity" in the real world, a line has a considerable number of properties. A linear feature on a map may be a geometric boundary, a sinuous river, a fractal or "area-like" coastline with indeterminate length, or a curve with a given expected smoothness such as a contour or fold line. This line is then geocoded and transformed into a spatial object called a string, a list of coordinates in some coordinate system which taken sequentially represent the entity inside a computer. Common-sense (and cartographic theory) tell us that the entity can then be symbolized and displayed accurately only at scales less than or equal to that of the source information. If the source of a coastline was a 1:12,000 air photo, for example, the line can be shown with confidence at a scale of 1:24,000 or 1:100,000.

At scales much smaller than that of the source entity, the problem becomes one of generalization. This problem is particularly well understood in cartography, and has been the subject of considerable research (McMaster and Shea, 1992; Buttenfield and McMaster, 1991). While applications of

"cartographic license" are occasionally necessary, (e.g. the movement or displacement of features, elimination or joining of islands, etc.) line generalization has traditionally focussed on reducing the number of points required to represent a line, with the constraint that the points continue to lie on the line, or be one of the original set of points (Douglas and Peucker, 1973).

Consider, however, the inverse transformation. Cartographic enhancement seeks to increase the number of points used to represent a string as a cartographic object. Enhancement can be thought of as the compliment to generalization (Clarke 1982). The goal of enhancement is to add detail to generalized maps and restore them to their original state. This may be nearly impossible where no high resolution variance data are available, but nevertheless enhancement can make cartographic lines more visually appealing for display purposes.

By parallel with generalization, we can state that enhancement can also take two forms. First, enhancement can simply densify the number of points given the existing line. An example of this type of representation would be the conversion of a line to a grid data structure or to Freeman codes. Sampling a line with a set of points which are equidistant along the line, separated with a fine spacing would be a similar example. The number of locations increases, though the line remains the same. Minor generalization will sometimes occur when points on the line are separated by at least half the grid spacing. Critical in this type of enhancement, termed ***emphatic enhancement***, is the use of the original source string as the source of information in the enhanced line. This can be thought of as having all new points lie on or very close to the original line.

Alternatively, we can apply a model to the line, and allow the model to generate points which are not constrained to the geometry of the original line. This type of enhancement is ***synthetic enhancement***, in that the character of the line is imposed by the model used to enhance the line. In computer graphics, it is commonplace to enhance a line by fitting Bezier curves, Bessel functions, or B-splines (Newman and Sproull, 1979). These are mathematical models of the variation expected between points along the line, and assume that "smoothness," or continuity of the second derivative of a line is important. These methods are recursive by segment, that is they are applied to one segment at a time in a pass along the length of the string. In cartography, this type of smoothing enhancement is commonplace in contouring systems, where a curved and regular smoothness is expected by the interpreter, whether or not it is a good model of the actual data. Only one major alternative to this method for line enhancement exists, the fractal line enhancement algorithm of Dutton (1981). This method is also recursive by segment.

DUTTON'S FRACTAL ENHANCEMENT METHOD

It appears that Dutton has been alone in developing a technique to enhance the appearance of line detail. Dutton's algorithm is based on the two related properties of fractals, self similarity and fractional dimensionality. "Self similarity means that a portion of an object when isolated and enlarged exhibits the same characteristic complexity as the object as a whole."(Dutton 1981, p. 24) Fractional dimension means "the Euclidean dimension that normally characterizes a form (1 for lines, 2 for areas, 3 for volumes)" (Dutton 1981, p. 24).

Dutton's algorithm simply selects the midpoint of two adjacent line segments and moves the vertex out at a specified angle (Buttenfield 1985). Several parameters control the displacement of a vertex. The first is the Sinuosity Dimension (SD). SD determines "the amount of waviness that the chains should possess after fractalization."(Dutton 1981, p. 26). The next parameter is the uniformity coefficient (UC). UC is the proportion of distance to displace each vertex toward the recomputed location. A UC of 1 is the maximum an angle can be moved. A UC of zero compensates for the effect of fractalization and the lines appear unaffected. UC can range to -1, though anything below zero moves the vertex in the opposite direction of the original angle's original position. The last two parameters are straightness (ST) and smoothness (SM). ST is the maximum length at which a line segment will be altered. If a segment is greater than ST it will not be altered. This factor prevents the modification of long straight segments such as state and county boundaries. The last step is smoothing. Smoothing is used to enhanced the appearance of the lines for display purposes. In addition, smoothing can prevent the instances where line concavities result in generating a line which intersects itself.

Dutton's algorithm was implemented by Armstrong and Hopkins when they altered the digital layout of a stream network through fractalization (Armstrong and Hopkins, 1983). The stream network is represented as a interconnecting chain of grid cell centroids, with each cell one square mile in size. When the chains were plotted several problems resulted due to the rectilinear nature of the data. The bulk of the problems were spikes and streams joining in grid cells further downstream. To solve these problems, the "chains were smoothed (using splines) before fractalizing."(Armstrong and Hopkins 1983) After initial smoothing, Dutton's algorithm was applied and the chains were again smoothed. This resolved the problems and created a more natural appearance for the stream network. Dutton's fractal line enhancement technique was empirically tested and shown to provide convincing enhanced lines.

Dutton discounted the use of single rather than recursive mathematical transformations for enhancement. "Digitized map data resemble fractals much more than they resemble (continuous) functions which mathematicians normally study. Although certain cartographic objects, including both boundaries and terrain, can be approximated using real functions, e.g. trigonometric series, the difficulty remains of representing nonperiodic map features as well as ones that are not single-valued, i.e., places where curves reverse direction" (Dutton 1981, p. 25). We have taken Dutton's statement as a challenge, and have implemented a Fourier Line Enhancement technique which is able to add detail to digital line data, and successfully deals with the problems which Dutton cites.

FOURIER LINE ENHANCEMENT

A new technique for line enhancement is presented here. The heritage of the method lies in image processing, shape analysis (Moellering and Rayner, 1979) and terrain analysis (Clarke, 1988; Pike, 1986). The mathematical model chosen to enhance the lines is that of Fourier analysis, which assumes that the two series of points (we treat x and y as independent series and conduct a separate one dimensional Fourier transform for each), can be abstracted as the sum of a set of periodic trigonometric functions.

The method works as follows. First, the input string is resampled into two series (x and y) which are functions of a counter n, where n represents equidistant steps along the string, starting at the initial point (which is retained), and incrementing in equal distances along the line. This is similar to the so-called walking dividers method for computing the fractal dimension (Lam and DeCola, 1993) in that the line is resampled by uniform steps of a pair of dividers set to some spacing. The spacing distance was user defined, but in all cases was some ratio of the minimum observed segment length.

Each of the two sets of independent coordinates is then subjected to a discrete Fourier transform. The discrete transform is necessary since the length of the series is unknown until computation. The Fourier transform assumes that the original series in x and y are a sample from a set of trigonometric functions over a finite harmonic range. The full series, unsampled, is given by a transformed set of coordinates (u, v), which can be computed at any interval over the length of the series. The computer code for the series is adapted from Davis (1973), and has been rewritten in C. The computer program computes for each series two floating point vectors of Fourier coefficients, spaced as spatial harmonics rather than distances.

$$u_i = \sum_{k=1}^{kmax} \left(A_k \cos\left(\frac{2k\pi x_i}{\lambda}\right) + B_k \sin\left(\frac{2k\pi x_i}{\lambda}\right)\right)$$

$$v_i = \sum_{k=1}^{kmax} \left(A_k \cos\left(\frac{2k\pi y_i}{\lambda}\right) + B_k \sin\left(\frac{2k\pi y_i}{\lambda}\right)\right)$$

The contribution of the harmonic number k can be computed from:

$$S^2_{x,k} = A^2_{x,k} + B^2_{x,k} \qquad S^2_{y,k} = A^2_{y,k} + B^2_{y,k}$$

$$P_{k(x,y)} = \frac{S^2_{x,k}}{\sum_{k=1}^{kmax} S^2_{x,k}} + \frac{S^2_{y,k}}{\sum_{k=1}^{kmax} S^2_{y,k}}$$

In brief, the x and y series are abstracted by a set of combined sine and cosine waves of different amplitudes, wavelengths and phase angles. These angles are computed for harmonics, where the first harmonic is the mean

of the series, and the second through kmax are of wavelength λ of length n/k. Thus the sixth harmonic has a wavelength of one sixth of the length of the series. The S values are similar to variances in least squares methods, and combine to give the proportion of total variance contributed by any given harmonic pair P. Typically, the bulk of the variance combines into a few harmonic pairs. Choosing these pairs only, zeroing out the remainder of the Fourier series, and then inverting the series (computing x and y from the given A and B's) is exactly equivalent to smoothing the series.

Enhancement of the series is now possible in either of two ways. First, the series can be oversampled into a large number of equidistant points, and then harmonics meeting a predetermined level of significance can be chosen for the inverse transformation. This inversion yields the same number of points as the resampling, which can then again be resampled by n-point elimination to give the desired level of detail.

Alternatively, the complete Fourier series can be modified by applying an arbitrary function to the series which increases the high frequency end of the spectrum. The inverse transform then produces an emphatic enhancement by introducing more high frequency into the line. This method has been implemented in a C program which modifies the computed Fourier coefficients according to the following formulae:

$$A_k = A_k(1 + \frac{k-1}{n-1}), k = 2...n$$

$$B_k = B_k(1 + \frac{k-1}{n-1}), k = 2...n$$

This enhancement successively modifies the Fourier coefficients by a scaling, increasing from zero at the first harmonic (the series mean), one at the second harmonic, to two at the highest frequency harmonic. Thus successively higher frequency harmonics are amplified, yet by sufficiently small amounts that the Gibbs phenomenon is avoided.

Both of the above methods were implemented, as computer programs in the C programming language and using data structures and utility programs from Clarke (1990). Results were computed (i) for the same lines which Dutton used in his work and (ii) for a detailed digital shoreline of Long Island, New York.

RESULTS

The results of the analysis of Dutton's original test line are summarized in Table 1. Several different runs of the Fourier analysis were performed on the line. The line was first transformed into the spatial frequency domain with the tolerance set to 0.0001 (tol=0.0001) of the total variance contributed to the series by any given harmonic. Any harmonic with an S value less than the tolerance was set to zero. The line was resampled to the minimum segment length (s = 1). The result of the resampling produced a line of 71 segments from a line of 12 segments, giving an emphatic enhancement of the original line. The mean segment length and standard deviation of the segment lengths were increased. When the sampling interval is shortened to 1/3 of the minimum segment length (s = 3), the standard deviation of the segment lengths increased, which may be an artifact of the Gibbs phenomenon (see Figure 1). As the tolerance is increased, fewer harmonics are used in the inverse transform. Consequently, the standard deviations increase, however, the lines are not being enhanced, but rather they are generalized due to fewer high frequencies remaining in the series. This generalization is not constrained by the geometry of the original line. Due to the very short nature of the line (12 segments) some obvious boundary effects occur at the ends. These could be alleviated by extending the lines, by forcing the first couple of segments to conform to the source line, or by truncating the line before the ends.

The highpass spatial frequency filter was applied to a line resampled by the minimum segment length with the tolerance set to zero. All frequencies were retained in the inverse transform. The result of the highpass filtering was no different from the transform using a tolerance of 0.0001 and an s of 1. The standard deviations of the segment lengths, the minimum, maximum, and mean segment lengths remained the same to two decimal places, however, upon visual interpretation (see Figure 2), there is a slight increase in the high frequency variation of the line. The effect is minimal. The result is comparable to an emphatic enhancement. This may be a result of the particular frequency distribution of the power spectrum. More variation in the "wiggliness" of the line was expected. This effect is explored further with a larger data set from a digitized portion of the coast of Long Island.

Table 1:

Enhancement	Number of Segments	Minimum Segment Length	Maximum Segment Length	Mean Segment Length	Standard Deviation of Segment Length
Dutton: Original Line	12	62.55	975.19	38.43	27.94
Dutton: Fractalized D=1.1	36	47.01	272.72	124.19	63.71
Dutton: Fractalized D=1.5	39	58.52	304.25	139.40	69.02
Dutton: D = 1.1 Smoothed	37	44.63	605.69	183.97	125.38
Dutton: D = 1.5 Smoothed	39	59.08	605.05	200.77	109.74
Fourier tol=0.0001 s = 1	71	20.53	107.01	67.81	31.36
Fourier tol=0.0001 s = 3	220	2.12	295.22	34.90	45.78
Fourier tol=0.002 s = 1	71	7.76	764.51	109.15	125.62
Fourier tol=0.002 s = 3	220	1.01	434.35	39.58	59.52
Fourier tol=0.003 s = 1	71	8.59	749.02	108.47	124.23
Fourier tol=0.003 s = 3	220	1.29	335.01	38.21	50.96
Highpass tol=0.0 s = 1	71	20.53	107.01	67.81	31.36

A section of a digitized map of Long Island was used to test the efficacy of the highpass algorithm on a real data set. The highpass algorithm was run with the same parameters as with the Dutton test line (tol=0.0, s=1). When the inverse transform was performed, the number of points in the line was significantly increased, and the segments became shorter and more similar. However, aesthetically, there was no visual difference between the original generalized line and the highpass filtered line, except for a slight increase in the roundedness of acute angles. This may be a result of the scale used for display, but as a result, another highpass algorithm was tested.

The second highpass algorithm was designed to increase the high frequency portion of the spectrum more than the first algorithm. The second algorithm divided the power spectrum in half. The low end was left unadjusted, but the magnitude of the high frequency portion of the spectrum was doubled. The result was a coastline that appeared very realistic (see Figure 3). The coastline had undulations, which appeared to be the result of mass wasting and wave action. This is to be expected because wave action is a circular motion, and the Fourier analysis uses a sinusoidal wave form as its model.

Table 2:

Enhancement	Number of Segments	Minimum Segment Length	Maximum Segment Length	Mean Segment Length	Standard Deviation of Segment Length
Long Island: Original Line	96	107.57	4272.91	807.97	626.50
LI Highpass tol=0.0 s = 1	696	73.53	127.68	107.51	1.98
LI Highpass2 tol=0.0 s = 1	696	12.37	582.87	143.38	297.35

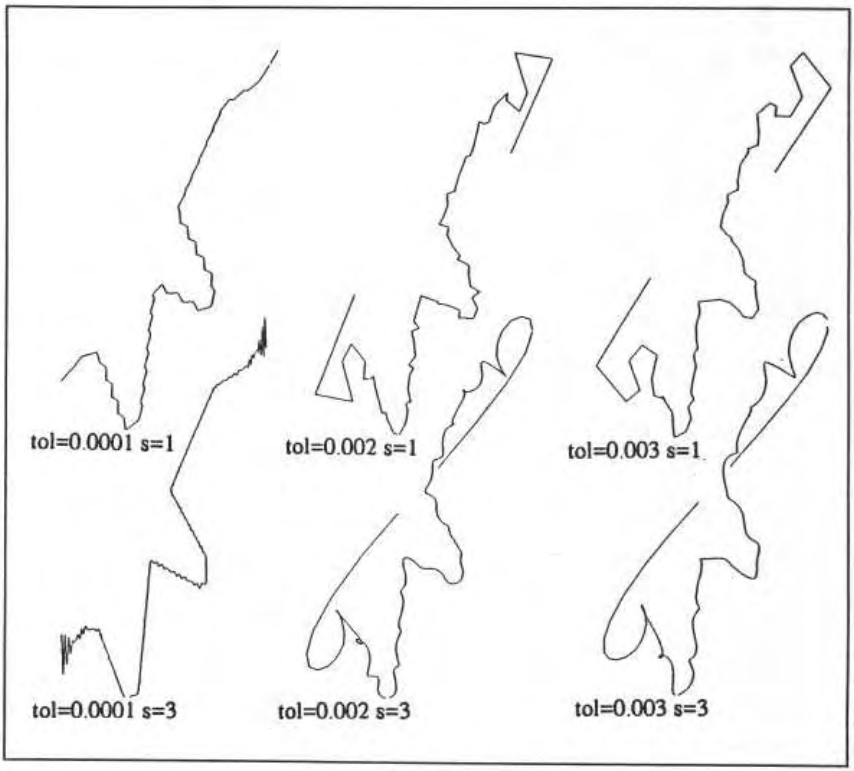

tol=0.0001 s=1
tol=0.002 s=1
tol=0.003 s=1
tol=0.0001 s=3
tol=0.002 s=3
tol=0.003 s=3

Figure 1

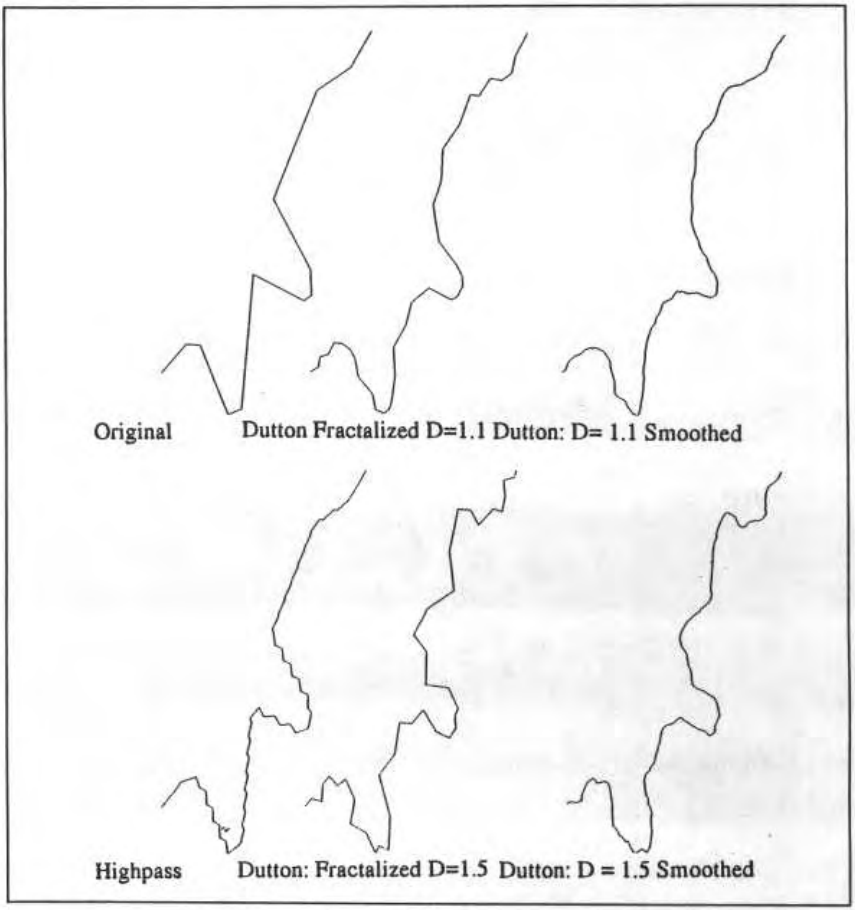

Original
Dutton Fractalized D=1.1
Dutton: D= 1.1 Smoothed
Highpass
Dutton: Fractalized D=1.5
Dutton: D = 1.5 Smoothed

Figure 2

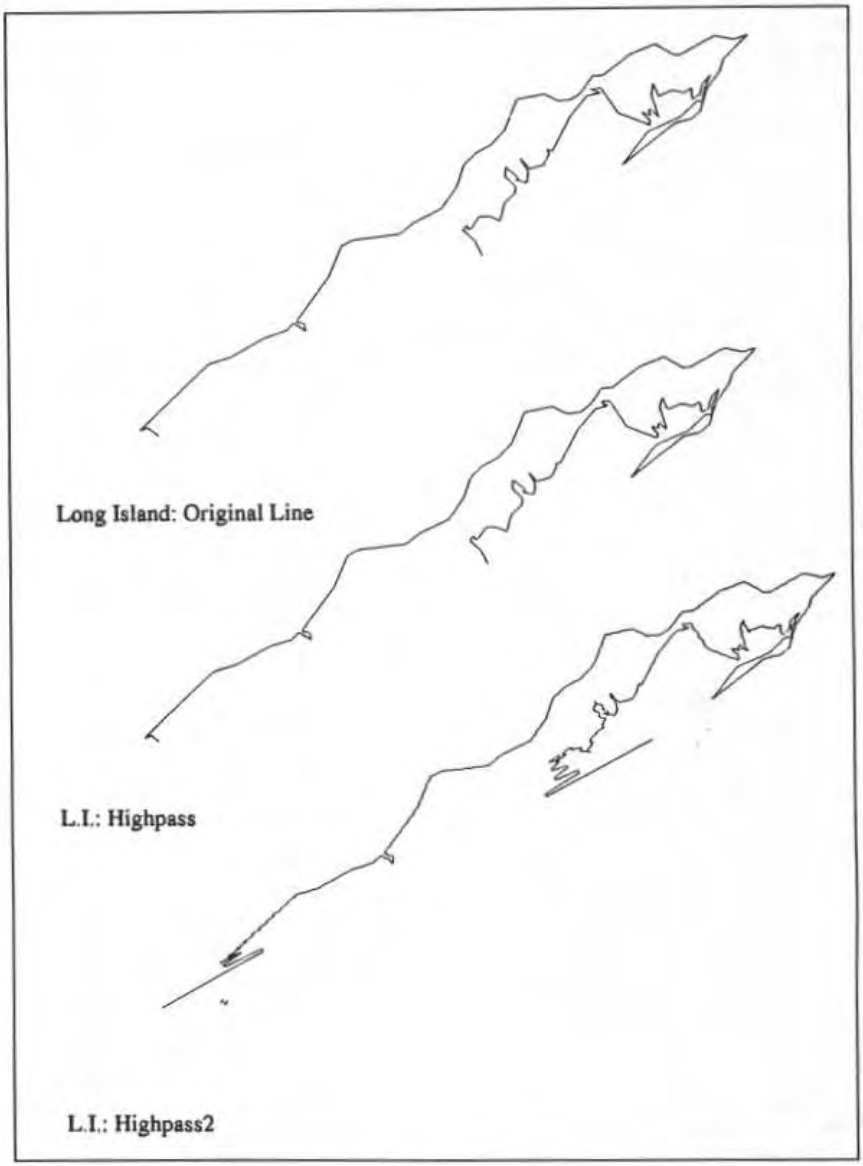


Figure 3

DISCUSSION

Like many algorithms, the Fourier enhancement method is controlled by a set of parameters. In our tests, we have found that three factors control the effectiveness of the technique and the character of the resultant enhancement. These are (i) the original distance used as a sampling interval to resample the points with equidistant spacing (ii) the amount of variance a harmonic should account for before it is included in the inverse Fourier transform, and (iii) the means of modification of the high frequency harmonics in the Fourier domain.

The choice of a proportion of the minimum segment distance in the line was made after several experiments with other measures, and was influenced by the relationship between our chosen mathematical model and the sampling theorem. Oversampling ensures that the enhanced line resembles the original line. We do not recommend sampling at distances greater than the minimum segment length, since the result is Fourier based generalization. This may be fine for smooth lines such as contours, but would not be good for coastlines. In addition, the choice of sampling interval determines the number of points in the analysis, the number of harmonics in the Fourier domain, and the number of points in the reconstituted enhancement. The use of oversampling followed by simple n-point elimination is seen as the most effective combination should a maximum number of points be desired in the final enhancement. Further research could include choosing sampling intervals which allow use of the Fast Fourier Transform, which would considerably increase computation speed.

The criterion for inclusion of a harmonic in the inverse Fourier transform was the amount of variance or "power" of the particular harmonic. Setting the inclusion tolerance to zero includes all of the harmonics, and in all test cases, reproduced the original line almost perfectly. This property of the Fourier transform is highly desirable, since the Fourier domain then contains the same amount of information as the spatial domain. Even complex lines are remarkably well represented by a limited set of harmonics, yet as the number of harmonics falls (the tolerance increases) the limitations of trigonometric series for modeling complex lines become more marked. Lines can self-overlap (not always seen as a problem for complex coastlines) which leads to topological discontinuities. In some cases, one or two harmonics make a large difference in the shape of the line. Leaving them out produces boundary effects and degenerate shapes for the lines (figure 1).

The means of modification of the series is another control variable. Tests showed that several modifications of the Fourier coefficients aimed at increasing amplitudes at the high frequency end of the spectrum produced satisfactory results. While the simple method tested for the Dutton data proved adequate, again further research on the effects of different modifications of the higher frequencies is called for. A method which produced different "flavors" of wiggliness at the highest frequencies under the control of a single variable would be most suitable for enhancement purposes.

CONCLUSION

We have successfully implemented and tested an alternative method for the synthetic enhancement of cartographic lines. The method has been empirically shown to give results comparable to those of Dutton's technique. Our method uses (i) an equidistant resampling of the source line to yield independent x and y series, (ii) a discrete Fourier transform of these two series, (iii) modification in the Fourier domain by either extraction of significant harmonics or deliberate amplification of the higher frequency harmonics and (iv) the equivalent inverse Fourier transform back into the spatial domain. Advantages of the method are that it is a single, rather than a piecewise recursive mathematical transformation; that it is analytic, i.e. the Fourier domain data have spatial meaning in terms of line shape, character, and fractal dimension; that it produces highly compressed Fourier domain representations of lines which can be retransformed to any scale, enhanced or generalized; and that it is computationally fast, especially in the Fourier to spatial domain transformation. In addition, lines with recognizably different character, from smooth to jagged, can be produced by the same method, including line with sharp directional discontinuities.

Cartographers have traditionally argued that enhancement, as the opposite of generalization, is undesirable since its accuracy is by definition unknown. Nevertheless, the history of manual and computer cartography is full of instances of the application of "cartographic license," or perhaps "digitizer's handshake," especially for linear features. Our method is objective and repeatable, and yields results which are esthetically what a map reader expects to see. Where enhanced linework is acceptable, the advantages of holding an intermediate scale database from which both higher resolution and smaller scale maps can be created quickly on demand by one method has some distinct advantages to the cartographic practitioner. As an analytical bridge between esthetics and mathematics, the method also has much to offer the analytical cartographer.

ACKNOWLEDGEMENT

The authors would like to thank Geoff Dutton, who generously provided his FORTRAN code for implementing the Dutton enhancement algorithm.

REFERENCES

Armstrong, M. P., Hopkins, L. D. (1983) "Fractal Enhancement for Thematic Display of Topologically Stored Data", *Proceedings, Auto-Carto Six*, vol. II, pp.309-318.

Buttenfield, B. (1985) "Treatment of the Cartographic Line", *Cartographica*, vol.22, no.2 pp. 1-26

Buttenfield, B. P. and McMaster, R. B. (Eds.) (1991) *Map Generalization: Making Rules for Knowledge Representation*, J. Wiley and Sons, New York, NY.

Clarke, K.C. (1982) "Geographic Enhancement of Choropleth Data", Ph. D. Dissertation, The University of Michigan, University Microfilms, Ann Arbor, MI.

Clarke, K. C. (1988) "Scale-based simulation of topographic relief", *The American Cartographer*, vol. 15, no. 2, pp. 173-181.

Clarke, K. C. (1990) *Analytical and Computer Cartography*, Prentice Hall, Englewood Cliffs, NJ.

Davis, J. C. (1973) *Statistics and Data Analysis in Geology*, J. Wiley, New York.

Douglas, D. H., and Peucker, T. H.(1973) "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature", *The Canadian Cartographer*, vol. 10, no. 2, pp. 112-122.

Dutton, G.H. (1981) "Fractal Enhancement of Cartographic Line Detail", American Cartographer, V.8, N.1, pp. 23-40.

Lam, N. and DeCola, L. (Eds.) *(1993) Fractals In Geography*, Englewood Cliffs, NJ, Prentice Hall.

McMaster, R. B. and K. S. Shea (1992) *Generalization in Digital Cartography*, Association of American Geographers Resource Publication Series, Washington, D.C.

Moellering, H. and Rayner, J. N. (1979) "Measurement of shape in geography and cartography", *Reports of the Numerical Cartography Laboratory, Ohio State University*, NSF Report # SOC77-11318.

Newman, W. M. and Sproull, R. F. (1979) *Principles of Interactive Computer Graphics*, McGraw-Hill, New York. 2nd. Ed.

Pike, R. J. (1986) "Variance Spectra of Representative 1:62,500 scale topographies: A terrestrial calibration for planetary roughness at 0.3 km to 7.0 km", *Lunar and Planetary Science*, XVII, pp. 668-669.

SAMPLING AND MAPPING HETEROGENEOUS SURFACES BY OPTIMAL TILING

Ferenc Csillag
Institute for Land Information Management, University of Toronto,
Erindale College, Mississauga, ONT, L5L 1C6
tel: 416-828-3862; fax: 416-828-5273; e-mail: fcs@geomancer.erin.utoronto.ca

Miklós Kertész, Ágnes Kummert
Research Institute for Soil Science and Agricultural Chemistry,
Hungarian Academy of Sciences, H-1022 Budapest, Herman Ottó u. 15.

ABSTRACT

A novel approach has been developed, implemented and tested to approximate, or to map, large heterogeneous surfaces with predefined accuracy and complexity. The methodology is based on tiling, the hierarchical decomposition of regular grid tessellations. The quadtree-construction is guided by a measure of local homogeneity and the predefined number of leaves, or level of accuracy. A modified Kullback-divergence is applied for the characterization of goodness of approximation. The procedure is aimed to find the quadtree-represented map of limited number of leaves which is the most similar to a given image in terms of divergence. Various statistical and computational advantages are demonstrated within the framework of spatial data processing and error handling in geographic analysis. This methodology minimizes information loss under constraints on size, shape and distribution of varying size mapping units and the residual heterogeneity is distributed over the map as uniformly as possible. As such, it formally defines a cost-versus-quality function in the conceptual framework of the cartographic uncertainty relationship. Our straightforward decomposition algorithm is found to be superior to other combinations od sampling and interpolation for mapping strategies. It is advantageous in cases when the spatial structure of the phenomenon to be mapped is not known, and should be applied when ancillary information (e.g., remotely sensed data) is available. The approach is illustrated by the SHEEP (Spatial Heterogeneity Examination and Evaluation Program), an environmental soil mapping project based on high-resolution satellite imagery of a salt-affected rangeland in Hortobágy, NE-Hungary.

INTRODUCTION

The spatial structure of phenomena, its relationship to sampling, mapping, resolution and accuracy, has been the focus of a wide array of geographic research (Moellering and Tobler 1972; Goodchild 1992). Geographic information systems (GIS) are being increasingly used to utilize these findings related to sampling design to perform analyses on spatial databases and to evaluate their quality (Burrough 1986). These advances,

however, are rarely applied simultaneously in reported case studies, even though conceptually and methodologically they often have common foundations (for example, spatial covariation may be utilized in interpolation but not in sampling design and regionalization, or quality assessment).

We consider the conceptual framework for database development from sampling to the measurement of accuracy, and explicitly formalize the cost-versus-quality function. Sampling strategies for mapping are always based on some preliminary knowledge and a priori assumptions about the phenomenon to be mapped. They are based on expertise in the field of application or on mathematical statistics or on both. The "goodness," or efficiency of sampling, in general, is dependent on the relationship between the cost of sampling and the quality of the final map product. Regardless of the actual circumstances, the goal of sampling is to collect "representative" samples (Webster and Oliver 1990). We need a means of sampling that will ensure appropriate information that can predict characteristics at locations where no samples were taken. In terms of characterization, our analysis and discussion will be confined to the task of predicting the local (expected) value of a variable, which is a quite widely explored problem in mapping[1] (Ripley 1981).

Our assumptions about the circumstances of database development are as follows:

- a fixed budget is given for sampling, for which we seek maximum accuracy (or conversely, an accuracy threshold is given for which minimum sampling effort should be determined);
- the spatial structure of the phenomenon to be mapped is not homogeneous (i.e., it varies over a range of scales); therefore, no a priori partitions can be defined; and
- some ancillary data sets are available whose spatial pattern is in close correspondence with the phenomenon to be mapped.

A CONSTRAINED OPTIMAL APPROACH TO MAPPING A LATTICE

We provide here a method for constrained optimal approximation of lattices. The mapping of a two-dimensional heterogeneous surface is treated with the following constraints: (1) a data set (**A**) is available on a lattice, which will be sampled and approximated by a map (**M**); (2) both **A** and **M** are two-dimensional discrete distributions; (3) the location of each datum on **M** corresponds to the location of a datum or data on **A**; and (4) **M** consists of a finite number of patches that are homogeneous inside, and the value associated with a patch approximates the value(s) of **A** at corresponding locations. No assumptions are made about the exact nature of the spatial statistics of the surface to be mapped (e.g., stationarity).

FROM MULTIPLE RESOLUTION TO VARYING RESOLUTION

Spatial pattern can play a significant role in the characterization of patches from sampling through interpolation to regionalization.

[1] For a review of sampling, resolution and accuracy see Csillag et al. (1993).

Understanding spatial pattern generally aims at the design of a sampling scheme, which is reasonable under certain statistical assumptions and models (Kashyap and Chellapa 1983, Kabos 1991), and for the application of "best" parameters defined by those models in interpolation and representation (Tobler 1979; Jeon and Landgrebe 1992). Advancements in computing and storage performance in GIS, paralleled with the apparent contradiction between finding the best resolution for regular samples and identifying (a priori) patches (partitions) for samples (Webster and Oliver 1990), has increased the popularity of mapping with multiple representation (Dutton 1984). Beside some storage- and processing-related technical issues, the identification and representation of mapping units (or area-classes; see Mark and Csillag 1989) have not been adequately addressed. Several soil or vegetation maps, whose patches often contain inclusions, can serve as simple illustrations: when a partition is optimal for the patches, it misses the inclusions, and when it is optimized for the inclusions, it becomes redundant for the patches (Csillag et al. 1992). Furthermore, it is prohibitive, because of size, to examine all possible partitions for optimization.

In the construction of databases, the hierarchy of resolutions in multiple representations based on uniform grids, a pyramid, offers the possibility of creating a GIS, which adjusts the level of detail to particular queries. Furthermore, it has become feasible to create varying resolution representations, of which quadtrees have become most well known and widely applied (Samet 1990). Beside storage and processing efficiency, the advantage of varying resolution representations is to ensure uniform distribution of accuracy (quality) over the entire data set. This would require criteria for creating quadtrees from pyramids.

We have developed and implemented a method (Kertész et al. 1993) to create a quadtree as a constrained optimal approximation of the lowest (most detailed) level of a pyramid. A quadtree can also be thought of as a spatial classification (partition on tiles) with the advantage that not only does each location belong to one and only one leaf, but the location, size and arrangement of all possible leaves is known a priori. Our method starts from the highest (least detailed) level (i.e., approximating the entire lattice by one value, its mean) and proceeds by straightforward decomposition. Because of the strong constraint on the shape, size, arrangement and potential number of patches on a map represented by a quadtree as a function of the number of levels (Samet 1990), with the application of an appropriate measure, the accuracy of all potential maps can be compared -- hence the process can be optimized.

DIVERGENCE MEASURES TO CHARACTERIZE DIFFERENCES IN SPATIAL PATTERN

To quantify the dissimilarity between the lattice and its (potentially varying resolution) map we apply a measure that (1) directly compares the lattice and the map and returns a scalar, (2) has a value independent of the size of the lattice, (3) is nonparametric, (4) is independent of the scale of data values (i.e., invariant to multiplication by a constant), and (5) provides the opportunity for additive application due to element-by-element computation (Figure 1).

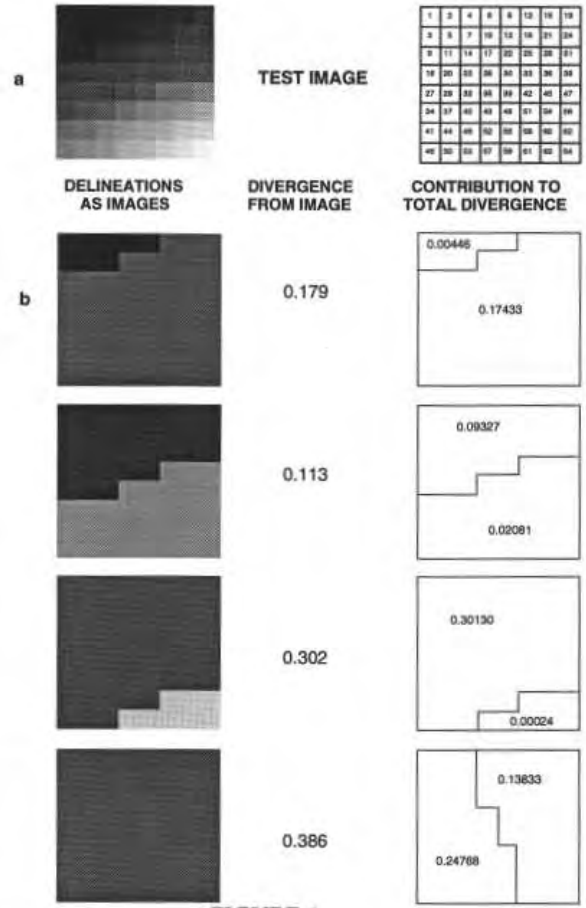


FIGURE 1.

Test data set (a) with four possible delineations with their corresponding total divergence and the contribution of the patches (b).

We have chosen Kullback-divergence (Csiszár 1975),

$$D_{Kullback}(p \mid q) = \sum_{i=1}^{I} \sum_{j=1}^{J} p_{ij} \log_2(p_{ij}/q_{ij}) \qquad (1)$$

where

$$\sum_{i=1}^{I} \sum_{j=1}^{J} p_{ij} = \sum_{i=1}^{I} \sum_{j=1}^{J} q_{ij} = 1, \qquad p_{ij} \geq 0,\ q_{ij} \geq 0$$

and

p and q are discrete spatial distributions of I by J elements;

because its usage does not require any limitations concerning the nature of

the distributions, and described its general properties and its relationship to Shannon's information formula, χ^2 elsewhere (Kertész et al. 1993). Its real advantage is that for any delineated patch on a map, one can compute the contribution of that particular patch to the total divergence:

$$D(patch_{grid} \mid patch_{map}) = \sum_{i}^{patch} \sum_{j} [grid_{ij}/SUM] \; log_2(grid_{ij}/map_{ij}) \qquad (2)$$

where

$grid_{ij}$ is the value for i,jth cell of the grid
map_{ij} the value for i,jth cell of the map
I and J are the side lengths of the grid to be mapped and the map in corresponding cells, respectively.

DECOMPOSITION PROCEDURE AND CRITERIA

Our method utilizes the advantage of varying resolution adjusted to spatial variability because it provides a rule for selecting the necessary spatial detail to represent each mapping unit with similar (internal) heterogeneity. In other words, it leads to a measure of local variability not weighted by area. Therefore, at very heterogeneous locations it will lead to smaller units (i.e., higher levels of the quadtree), whereas at relatively homogeneous locations it will decompose the grid into larger tiles (i.e., lower levels of the quadtree).

The decomposition algorithm can be characterized by two features: the cutting rule and the stopping rule. In this paper we use "maximum local divergence" as the cutting rule, and "total number of mapping units" as the stopping rule. Further options will be discussed in the final section. The first rule means that the quadtree leaf is cut into four quadrants whose local divergence is the maximum among all existing leaves, while the second rule stops the decomposition at a threshold predefined by the number of leaves.

The approach embedded in this method avoids the major conceptual problems of spatial pattern analysis and measurement of accuracy tied to the existence and knowledge of the spatial covariance function. In several, primarily environmental, mapping tasks the uncertainty in the delineation of mapping (and sampling) units is intolerably high, especially when local factors control the landscape. Kertész et al. (1993), for example, characterized salt-affected, semiarid rangelands exhibiting scale-invariant patterns; certain (mostly transitional) patches occurred over several hectares as well as over a few square centimeters. The measurement of accuracy in databases constructed for such areas will heavily depend on the validity of statistical assumptions (Goodchild and Gopal 1989; Csillag 1991). Our method not only does not require strong assumptions about the spatial statistics of the lattice but (1) optimizes local accuracy of sampling and mapping units under a global threshold, (2) explicitly links accuracy to the number of mapping units (and vice versa) and (3) when those strong assumptions are valid, leads to identical results.

MAPPING WITH ANCILLARY DATA: SAMPLING QUADTREES

We demonstrate the characteristics of the method described in the previous section from a database development perspective (i.e., as if one had to actually sample a surface and approximate it on a lattice), with several illustrative examples. For comparison, besides sampling based on quadtrees, equal number of samples will be taken by random and regular (square) design. Reconstruction will be performed by Thiessen-polygonization, inverse squared Euclidean distance, and kriging to test the robustness of the sampling method over various levels of sophistication in interpolation.

All procedures are carried out on a data set, with spatial characteristics illustrated in Figure 2, taken from a satellite image used for database design and development in SHEEP (Spatial Heterogeneity Examination and Evaluation Program), an environmental rangeland degradation mapping project in East Hungary (Tóth et al. 1991; Kertész et al. 1993).

Selecting optimal samples by the proposed decomposition method requires some information in the form of a lattice about the surface to be sampled and mapped. At first, the entire lattice is approximated by its (global) mean (the root of the quadtree), and its Kullback-divergence is measured from the actual distribution. Then, the lattice is approximated by four quadrants (level 1 on the quadtree), and for each leaf their contribution to the total Kullback-divergence is computed according to Equation (2). If the threshold in the number of samples or in total Kullback-divergence has not been met, decomposition is continued by cutting the leaf with the highest contribution (i.e., the most heterogeneous one). Thus, at each step, the number of (potential) sampling units increases by three, and the decomposition proceeds least intensively over homogeneous areas.

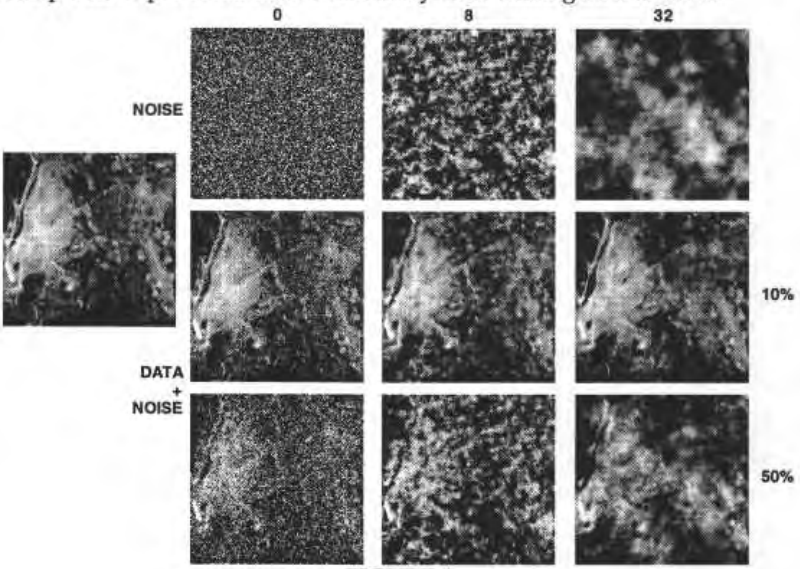


FIGURE 2.
The original data set (left), some of the noise fields (top) and their combinations (bottom).

To compare the efficiency of the different sampling designs we reconstructed the 128-by-128 data sets and measured their Kullback-divergence from the original data. For realistic illustration we set the number of samples to 256. Three interpolations (quadtree leaves, Thiessen-polygonization, and inverse squared Euclidean distance) are summarized for the sampling quadtrees (256 leaves), and three interpolations (kriging, Thiessen-polygonization, and inverse squared Euclidean distance) are shown for the regular and random sampling. The numerical results are summarized in Table 1.

Table 1.
Accuracy of approximations (D_K*10^4 to the original data based on 256 samples) by sampling design (Q=quadtree, G=regular grid, R=random) and interpolation (QTR=sampling quadtree, THI=Thiessen-polygonization, DI2=inverse square distance, KRI=kriging).

INTERPOLATION	QTR	THI	DI2	KRI
SAMPLING				
Q	8.937	16.864	11.094	---
G	---	22.864	17.550	17.646
R	---	25.545	19.824	18.468

The sampling quadtrees lead to approximately half, or less, the Kullback-divergence than any other sampling and interpolation method, because they not only are more sensitive to local variability but also are obtained by using all data from the approximated distribution.[2] In addition, samples selected by this method carry over so much information about the distribution of variance in the data set that they systematically result in significantly smaller Kullback-divergence than the other sampling methods over all interpolation techniques. Furthermore, sampling based on quadtrees and interpolation by inverse squared Euclidean distance is, by far, the superior method among those tested.

SAMPLING AND RECONSTRUCTION: NOISY DATA

In general, at best, one can assume to have a data set that corresponds to the phenomenon to be mapped but contains a certain amount of noise. This available data set may be remotely sensed data, as in the illustrative example, or it can be any existing data set (on a lattice) in a database. In practice, these are exactly the sources of sampling design and database development. Therefore, it is important to examine the proportion and spatial structures of noise and its effects on the accuracy of sampling and reconstruction.

We generated noise fields with five different levels of spatial autocorrelation (correlation distance or range = 0, 4, 8, 16, 32 cell units)[3] and mixed them with 10%, and 50% weights to the original data sets (Figure 2),

[2] These results refer to the ideal situation of when the data set to be approximated in our database is entirely well known. The Kullback-divergences of the Q_QTR (sampling quadtree with mean values assigned to the leaves) method can be interpreted, therefore, as measures of the cost of data compression.

[3] Gerard Heuvelink kindly made his software available (Heuvelink 1982).

preserving the original mean and variance. The Kullback-divergences between the noisy and original data sets are summarized in Table 2. It helps to "scale" Kullback-divergence to certain amounts of noise, which reveals that while there is a strong relationship between the amount of noise and Kullback-divergence, it does not change significantly with its correlation length.

Table 2.
Kullback-divergences between the noisy and the original data sets (D_K*10^4)

	CORRELATION LENGTH				
	0	4	8	16	32
NOISE-LEVEL					
10%	4.782	4.802	4.856	5.128	4.912
50%	26.234	26.200	27.296	30.537	28.004

We examine the effects of noise by designing the sampling quadtree on the noisy data, and approximate the original one. Numerical results indicate that the longer range noise is added (i.e., the smoother the ancillary data set becomes), the better the approximation is to the ancillary data, and the accuracy of reconstructing the original data decreases. At 10% random noise, the 256 samples taken from the noisy ancillary data set approximate the original data almost as well as if samples were taken from the original ($9.532*10^{-4}$ versus $8.937*10^{-4}$ for tiling reconstruction).

Reconstructions based on sampling quadtrees with 256 samples using inverse squared Euclidean distance interpolation (providing the best results among the methods tested) outperform reconstructions based on random and regular square sampling using the same interpolation, regardless of the amount and spatial structure of noise (Table 3). The stronger the pattern, the more sampling quadtrees provide advantage (Figure 3). At low (10%) noise levels, consistently over all noise structures, reconstructions based on sampling quadtrees lead to approximately one-third less Kullback-divergence than reconstructions based on other sampling methods. All accuracies slightly increase when approximating original data.

Table 3.
Kullback-divergences (D_K*10^4) between reconstructions, the original data set and noisy data sets; 256 sampling locations determined on noisy data by sampling quadtree (Q), regular square grid (G) and random (R); samples taken from noisy data and inverse squared Euclidean distance interpolation.

	CORRELATION LENGTH				
	0	4	8	16	32
10%					
(sampled vs. sampled)					
Q	16.781	16.225	15.653	12.096	11.363
G	22.704	21.909	19.573	17.402	16.321
R	23.786	23.651	21.095	19.308	19.092
(sampled vs. original)					
Q	12.937	13.321	14.532	14.224	14.723

G	19.071	19.511	20.037	20.248	21.395
R	20.889	21.992	21.240	22.631	22.416
50%					
(sampled vs. sampled)					
Q	39.874	34.879	27.003	15.906	9.383
G	40.994	36.858	28.183	17.048	12.071
R	39.921	37.998	30.225	19.777	16.476
(sampled vs. original)					
Q	25.756	26.478	31.245	32.198	34.615
G	26.779	28.618	32.015	34.518	37.932
R	27.742	31.464	31.392	38.259	38.713

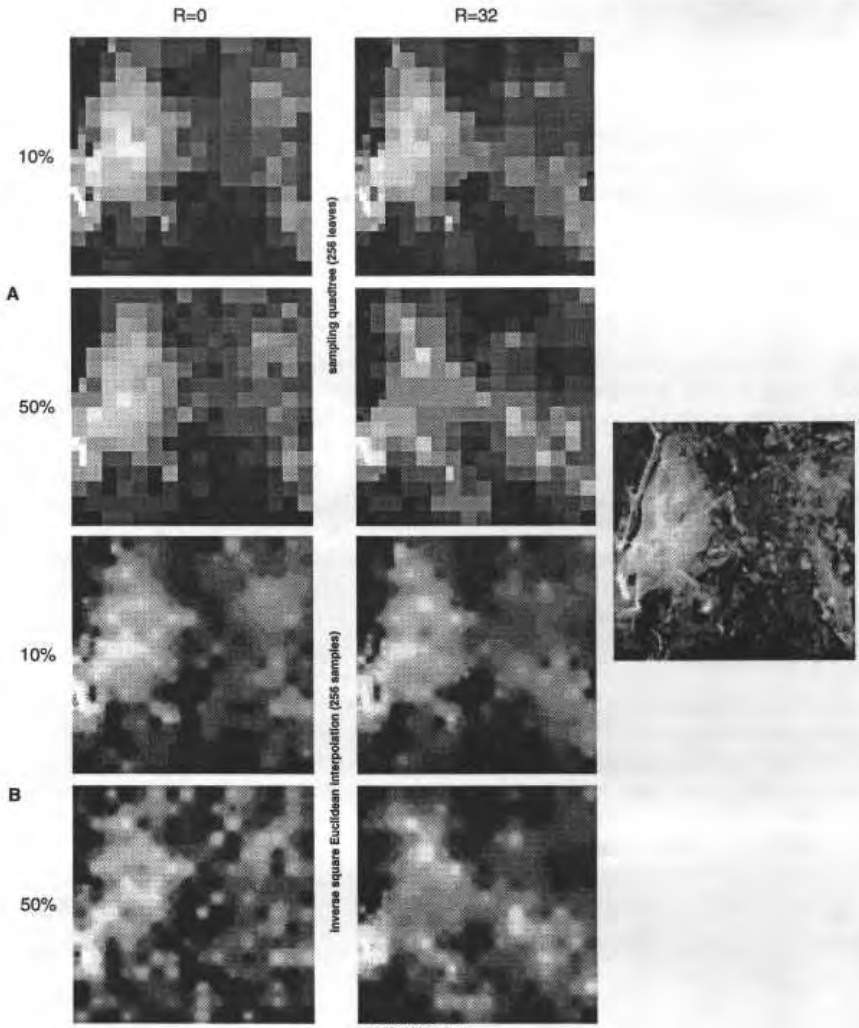


FIGURE 3.

Reconstructions based on 256 samples under various amounts and spatial structure of noise: sampling quadtree (A), inverse square distance (B).

At high (50%) noise levels the differences are more related to the spatial structure of noise. Whereas the Kullback-divergence of 50% noisy data from the original is about six times that of 10% noisy data (Table 2), the Kullback-divergences of reconstructions increase by a factor of only three. Reconstruction by inverse squared Euclidean distance interpolation based on 256 samples, selected by sampling quadtrees from the noisy ancillary data, approximate the original data set comparably than 256 samples selected by other sampling methods from the original data set (e.g., D_K increases from $12.937*10^{-4}$ to $14.723*10^{-4}$ as the correlation length of noise increases; these values are below the ones obtained for regular square sampling [G, $17.550*10^{-4}$] or random sampling [R, $19.824*10^{-4}$] of the original data and significantly below the D_K values between reconstructions and the noisy data set).

CONCLUDING REMARKS

The approximation or mapping procedure described above is a part of a larger project aimed to develop optimal resolution mapping for heterogeneous landscapes, and salt-affected grasslands in particular. The optimization is carried out by locally adjusting (changing) the spatial resolution of the map so that it conveys maximum information for the user with given (predefined) number of patches. Hence, it is a sampling effort constrained optimization.

The sampling quadtrees are computed controlling accuracy by Kullback-divergence, an information theoretical measure to characterize spatial pattern. Several modifications of the current algorithm (with the cutting rule tied to maximum contribution to total Kullback-divergence and straightforward decomposition) are under investigation, as well as are extensions to other, more flexible, hierarchical data structures and links to efficient computations of spatial statistical characteristics based on tile-size distributions (Kummert et al. 1992).

We evaluated the performance of the proposed method under various levels and different spatial structures of noise and compared the results with other (regular square and random) sampling. Reconstructions of two-dimensional distributions on a regular lattice based on sampling quadtrees outperform other sampling designs. The more heterogeneous the surface to be mapped and the fewer (realistically limited number of) samples taken, the more benefit can be gained.

This study provided the foundations for the sampling and mapping procedures of the SHEEP project in northeast Hungary. Further studies of the efficiency and robustness of this and related methods (e.g., by pruning the quadtree and/or formalizing the correspondence among mapped variables) will be evaluated and tested in several other test sites.

ACKNOWLEDGMENT

This research was supported under Grant No. DHR-5600-G-00-1055-00, Program in Science and Technology Cooperation, Office of the Science Advisor, U.S. Agency for International Development.

REFERENCES

BURROUGH, P. A. 1986. Principles of Geographical Information Systems. (Oxford: Clarendon Press).

CSILLAG, F. 1991. Resolution revisited. Proceedings of AutoCarto-10. (Bethesda: American Society of Photogrammetry and Remote Sensing/ American Congress on Surveying and Mapping), pp. 15-29.

CSILLAG, F., KERTÉSZ, M., AND KUMMERT, Á. 1992. Resolution, accuracy and attributes: Approaches for environmental geographical information systems. Computers, Environment and Urban Systems **16**: 289-297.

CSISZÁR, I. 1975. I-divergence geometry of probability distributions. Annals of Probability. **3**: 146-158.

DUTTON, G. 1984. Geodesic modeling of planetary relief. Cartographica **21**: 188-207.

GOODCHILD, M. F. 1992. Geographical information science. Int. J. Geographical Information Systems **6**: 31-46.

GOODCHILD, M.F. and GOPAL, S. 1989. Accuracy of spatial databases. (London: Taylor & Francis).

HEUVELINK, G. 1992. An iterative method for multi-dimensional simulation with nearest neighbour models. In P. A. Dowd and J. J. Royer (Eds.), 2nd CODATA Conference on Geomathematics and Geostatistics, **31**: 51-57, (Nancy: Science de la Terre).

JEON, B. AND LANDGREBE, D.A. 1992. Classification with spatio-temporal interpixel class dependency contexts. IEEE T. on Geoscience and Remote Sensing **30**: 663-672.

KABOS, S. 1991. Spatial statistics. (Budapest: Social Science Information Center, in Hungarian).

KASHYAP, R.L. AND CHELLAPA, R. 1983. Estimation of choice of neighbors in spatial-interaction models of images. IEEE T. on Information Theory. **IT-29**: 60-72.

KERTÉSZ, M., CSILLAG, F. AND KUMMERT, Á. 1993. Mapping heterogeneous images by optimal tiling. (manuscript) submitted to the Int. J. Remote Sensing.

KUMMERT, Á., KERTÉSZ, M., CSILLAG, F. AND KABOS, S. 1992. Dirty quadtrees: pruning and related regular decompositions for maps with predefined accuracy. Technical Report, (Budapest: Research Institute for Soil Science), p. 71.

MARK, D. M. AND CSILLAG, F. 1989. The nature of boundaries on 'area-class' maps. Cartographica **26**: 65-79.

MOELLERING, H. AND TOBLER, W. 1972. Geographical variances. Geographical Analysis **4**: 34-50.

RIPLEY, B. D. 1981. Spatial statistics. (New York: J. Wiley & Sons).

SAMET, H. 1990. Applications of spatial data structures. (Reading: Addison-Wesley).

TOBLER, W. R. 1979. Lattice tuning. Geographical Analysis **11**:36-44.

TÓTH, T., CSILLAG, F., BIEHL, L. L., AND MICHÉLI, E. 1991. Characterization of semivegetated salt-affected soils by means of field remote sensing. Remote Sensing of Environment. **37**: 167-180.

WEBSTER, R. AND OLIVER, M. 1990. Statistical methods in soil and land resource survey. (Oxford: Oxford University Press).

VIRTUAL DATA SET – AN APPROACH FOR THE INTEGRATION OF INCOMPATIBLE DATA

Eva-Maria Stephan, Andrej Vckovski and Felix Bucher
Department of Geography
University of Zurich
Winterthurerstr. 190
CH-8057 Zurich, Switzerland

ABSTRACT

Data integration within GIS is made difficult by the incompatibility of source data. Here both, traditional approaches are discussed and an enhanced strategy for data integration is proposed. The concept of a virtual data set is presented, which allows more flexible and – due to quality information – more reliable integrated analysis. As a conclusion implementation issues are discussed, including the benefits of object-oriented techniques, comprehensive data quality modelling and visualization.

INTRODUCTION

This paper focuses on strategies for data integration. It presents a conceptual framework for data integration, which is an aspect of a multi-year program supported by the Swiss National Science Foundation, dealing with climate change, its impacts on ecosystems in alpine regions, and its perception by human beings. Investigating spatial variations of climate change and its impacts on ecosystems requires the use of models where information of various kind of GIS databases are interrelated. This paper first discusses the mutual relationships between GIS, data integration and environmental data modelling and analysis. It then shows current strategies for data integration with a subsequent evaluation of the inherent problems. Finally we present and discuss an alternate concepts for successful data integration.

DATA INTEGRATION AND GIS

In the past decades the use of computer-based means for processing of spatial information has significantly grown in importance. Particularly the development of Geographical Information Systems (GIS) has contributed to this evolution and, as a consequence, has expanded the potential for the analysis of spatial processes and patterns. At the same time data production has grown enormously, a phenomenon which is sometimes called the *data revolution*. Data is now drawn from various sources, gathered with different methods and produced by different organizations. For these reasons data are often not directly comparable in an integrated analysis with respect to their spatio-temporal references and attribute characteristics. The new facilities in spatial data processing on the one hand and the effects of data revolution on the other hand have accentuated the *data integration problem*. Data integration can be defined as the process *„of making different data sets compatible with each other, so that they can reasonably be displayed on the same map and so that their relationships can sensibly be analysed"* (Rhind et al., 1984).

GIS link together diverse types of information drawn from a variety of sources. Thus information can be derived *„to which no one had access before, and* [GIS] *places old information in a new context"* (Dangermond, 1989, p. 25). In fact, the ability of GIS to integrate diverse information is frequently cited as its major defining attribute and as its major source of power and flexibility in meeting user

needs (Maguire, 1991). Data integration facilitates more accurate analysis of spatial processes and patterns and encourages to use interdisciplinary thinking for geographical problem solving. Finally, data integration is the most important assumption for GIS to meet the expectations as a tool for decision support for planning tasks.

Data Integration in the Context of Environmental Information

In general, the investigation of natural phenomena is a highly interdisciplinary task. Particularly the interaction and dynamics of specific natural processes is not yet well understood and is of great interest in current research. Therefore, data integration is of special importance in the field of environmental data analysis. One can assume that the more interdisciplinary an analysis is, the more likely data integration will be a problem. Beyond that, many spatial phenomena are either difficult or expensive to measure or to observe, requiring their estimation from cheaper or more readily available sources. The substitution of such spatial phenomena, sometimes referred to as the derivation of secondary data, has become increasingly relevant as an application in the field of GIS. Examples are the calculation of the amount of soil erosion by means of the Universal Soil Loss Equation (USLE) or the simulation of the spatial distribution of vegetation patterns based on a model which takes topographic, climatic and edaphic factors into consideration.

Heterogeneity as a Bottleneck for Data Integration

Geographical entities are by definition described by their spatial, temporal and thematic dimension. At the point when different data sets enter an integrated analysis, one is often confronted with the problem that data have different characteristics according to these dimensions. To point out different characteristics between data sets the term *inconsistency*[1] has been established in the field of GIS.

Many of the problems of heterogeneity are a consequence of the fact that data are an abstraction of reality. Depending on the degree of abstraction and on the conceptual model for the transformation of reality into a *data set*, these characteristics can vary quite strongly. Beyond that, heterogeneity between data sets is also being introduced by different data gathering strategies, previous preprocessing steps and a lack of standardization.

Data Integration and Data Quality

The difficulties in data integration that accrue from heterogeneity are often reinforced by the uncertainty that is inherent to the data. Consequently, the process of data integration should strictly include data quality assessment for the resulting data set. The effects of uncertainty in individual maps on data integration is the subject of extensive research by Veregin (1989), Chrisman (1989), Maffini et al. (1989) and Lodwick (1989), among others.

In fact, environmental data are particularly affected by uncertainty. There are many reasons for that, including:

- Thematic surfaces of environmental characteristics may not be directly visible and therefore may not be verified at a reasonable expense.
- Many natural phenomena undergo continuous change (e.g. coastlines).
- Some natural phenomena cause problems because their realizations cannot be distinguished clearly and have transitional zones (e.g. vegetation).

---

1. Actually, the term 'inconsistency' can be quite confusing. Inconsistency has both the meaning of *not in agreement* and *to be contradictory*. In the present context, inconsistency refers exclusively to its first meaning. To avoid confusion, we relate here heterogeneity and heterogeneous data sets, respectively, to inconsistency.

- Due to the high costs of data gathering sample sizes of environmental data sets are normally much smaller than in other fields (e.g. terrain elevations for the generation of a Digital Terrain Model).
- Some of the environmental data cannot be measured directly in the field and have to be subsequently analysed with physical or chemical laboratory methods.
- Natural phenomena often have not negligible non-deterministic variability.
- Local deterministic variations as well as intrinsic variability of the data often cannot be exhaustively captured with the samples drawn.

OPERATIONAL DATA INTEGRATION

Traditional Approaches

Traditional approaches consider data integration as a two-step procedure (Shepherd, 1991): The first step tries to reconcile heterogeneous data sets to ensure their comparability. The second step involves the use of appropriate procedures to interrelate consistent data in a way that they meet the needs of a particular application. The reason for this two-step approach is that heterogeneity exists between the different data sets according to the intended application. In an *operational* system these two steps are separated from each other. In order to simplify future applications, the data sets are transformed into a pre-specified, *common format* when entering the system, such that comparability is ensured (see also figure 2). Ideally, the common format should be based on requirements of future analyses, however those can hardly be estimated exhaustively. Unfortunately, its specification usually is restricted by non-context-specific criteria like economic, hard- and software, and data limitations.

In the past few years, some broad strategies for specifying a common format have been adopted. In some applications, comparability is achieved by reducing diverse spatial or attribute data to some lowest common denominator representation. For example, all attribute data may be down-scaled to nominal information, or all spatial data may be converted into a grid of coarse resolution. Other applications may not accept the reduction of the variability in the source information and achieve comparability by transforming data sets according to the data set with the highest resolution. Other approaches include the conversion of all source data into a single target version, as in the integration of multiple data base schemata (Nyerges, 1989) or, the conversion to one single data model, as is the case of vector-only or raster-only GIS (Piwowar et al., 1990).

In any case one must be aware, that such transformations on the original data introduce further uncertainty.

Problems and Improvements

The previous section has given an overview of current approaches for solving problems of data integration. Regardless of the fact that these approaches are widely accepted and applied, we are of the opinion that they need further improvement.

Data integration usually does not include data quality assessment: The procedures for reconciling heterogeneous data sets are performed by means of traditional (mechanistic) transformations (e.g. interpolation, scale and projection transformations) and result in spatio-temporal references and attribute characteristics that are only apparently equivalent. These transformations all have in common that they involve prediction of attribute data at predefined locations and points in time. Because prediction generally introduces further uncertainty, data quality assessment and error propagation methods must be included to evaluate the reliability of the resulting data set and its limitations for a particular applica-

tion. Furthermore, data quality information of component data sets is a prerequisite for the estimation and monitoring of the effects of uncertainty on integrated data sets. Openshaw (1989, p. 263) notices that *„the effects of combining data characterised by different levels of error and uncertainty need to be identifiable in the final outputs"*.

Data should always be explained by means of meta information: Meta information improve the data integration process at various stages. However, a highly structured format is necessary for operational use. Burrough (1991, p. 172) notices that *„formalization of the knowledge that we already have and putting that in a knowledge base next to a GIS would help the user choose the best set of procedures and tools to solve the problem within the constraints of data, data quality, cost and accuracy."* Ideally, meta information should include declarations about: (a) the process of which the data set is a realization; (b) the conditions during data capture; (c) former applications and preprocessing of the data (the so-called lineage or history of data); (d) Characteristics of the present format; (e) Quality and reliability (based on the information of points a-c) and limitations for specific applications. Beyond that, meta information even should include specifications of functionality to avoid inappropriate user actions with the data.

The specification of the common format is rigid: To avoid future problems of heterogeneity, GIS systems and data bases are often designed so that they store all data sets in a common format to ensure their comparability. Presumably the (rigid) common format does not meet the requirements of all future applications. As a consequence, additional transformations of the data are likely to be needed, which will introduce further uncertainty. This is especially fatal when expert knowledge would be needed for the transformations or predictions applied, but such information is not stored with the data.

It is difficult or even impossible to avoid these problems, since the specification of the common format is restricted by non-context-specific limitations. Additionally, the full adequacy of a given common format for all future applications can hardly be achieved.

A MORE FLEXIBLE APPROACH TO DATA INTEGRATION

Definition of Heterogeneity and Prediction

In the previous section heterogeneity was introduced as a term describing the inconsistency (or incompatibility) of two or more data sets. This incompatibility needs to be defined accurately to point out its significance to the problem of data integration. We start out with a definition of the term heterogeneity with respect to data integration:

> *Data sets to be combined are called heterogeneous (with respect to the specific application) if some data values need to be predicted before performing the integrating operation.*

The operation thus requires data that are not present in the original data sets and implies the use of prediction methods to derive the missing information. These methods most often are needed to predict attribute values at spatial locations and/or points in time using the data available in the original data sets[1] (e.g. extra- or interpolation methods). This definition of heterogeneity also includes other types of predictors, which sometimes may be degenerate in the sense that predicted values are analytical functions of the input values. A simple example of such a degenerate predictor would be a method that predicts temperature values

1. Of course, it is possible, and often desirable, to include additional data sets to improve predictions. We refer to such data sets as *original data sets* as well.

in degrees of Fahrenheit given the temperature in centigrade or the transformation from one coordinate system to another. While usual predictors increase uncertainty in the predicted data, there are predictors which leave the uncertainty unchanged or even reduce it.

This extension of the term *prediction* to analytically deducible values allows the use of the above definition of heterogeneity in a broader sense when discussing integration problems.

The Idea of a Virtual Data Set

In this section we present an approach that can be used in an operational system to overcome heterogeneity and to better encapsulate the expert knowledge within the data set. This concept is termed *virtual data set*. It is a proposal for a more generic approach to data integration.

The basic idea of the virtual data set is the *extension of a data set with methods to provide any derivable or predictable information*. Instead of transforming original data to a standard format and storing them, the *original data are enhanced with persistent methods that only will be executed upon request*.

As the name indicates, a virtual data set contains virtual data. Virtual data is information that is not physically present. That is, it is not persistent[1]. This data is computed on request at run time.

As outlined before the traditional approach solves the problem of heterogeneity using a common format for the data. Instead of transforming the data to that common format the virtual data sets include methods for such transformations (which are predictions in our sense) together with the original (unchanged) data. Neglecting implementation and performance details, those two approaches are equivalent as long as the application of the prediction methods are transparent to the user. The second approach, however, can easily be enhanced to be more efficient. Once the transformation or prediction methods for getting the data into the common format are known, it is often easy to define methods that transform the data into yet another format. Suppose a transformation exists, that interpolates an original data set in a grid of 1 km resolution. It will not be very difficult to change this interpolation method so that it will produce data on a 0.9 km resolution grid instead. It might even be possible to parametrize the method enabling it to supply data in any parameter-dependent resolution. The step from specialized to more general predictors is often small. Anyway, a representation consisting of the original data together with prediction methods always contains equivalent or more information than the transformed data itself.

Once the application of the prediction methods is fully transparent[2] to the user, it is preferable to enhance an existing data set with prediction methods instead of transforming it using those methods. It is important to note, that these methods are designed to provide quality information for each predicted value. It would be even favourable to have the quality information being an inherent part of both, virtual and original values.

Figure 1 shows a schematic view of an original data set, its transformation according to a common format and its enhancement to a virtual data set. The data set shows the spatial distribution of a certain phenomenon (attribute A) at given locations (s_1, s_2, s_3). The analysis requires an additional attribute B which can be derived from A. The common format specifies the locations s_a, ..., s_f where the attributes are interpolated. The virtual data set is equipped with methods to interpolate A at any location, to derive B from A and to transform spatial references between different coordinate systems.

1. We refer to persistent data if they are available on secondary (external) storage.
2. It should make no difference whether the user accesses data really available in the data set or virtual data that has to be computed using the prediction methods first.

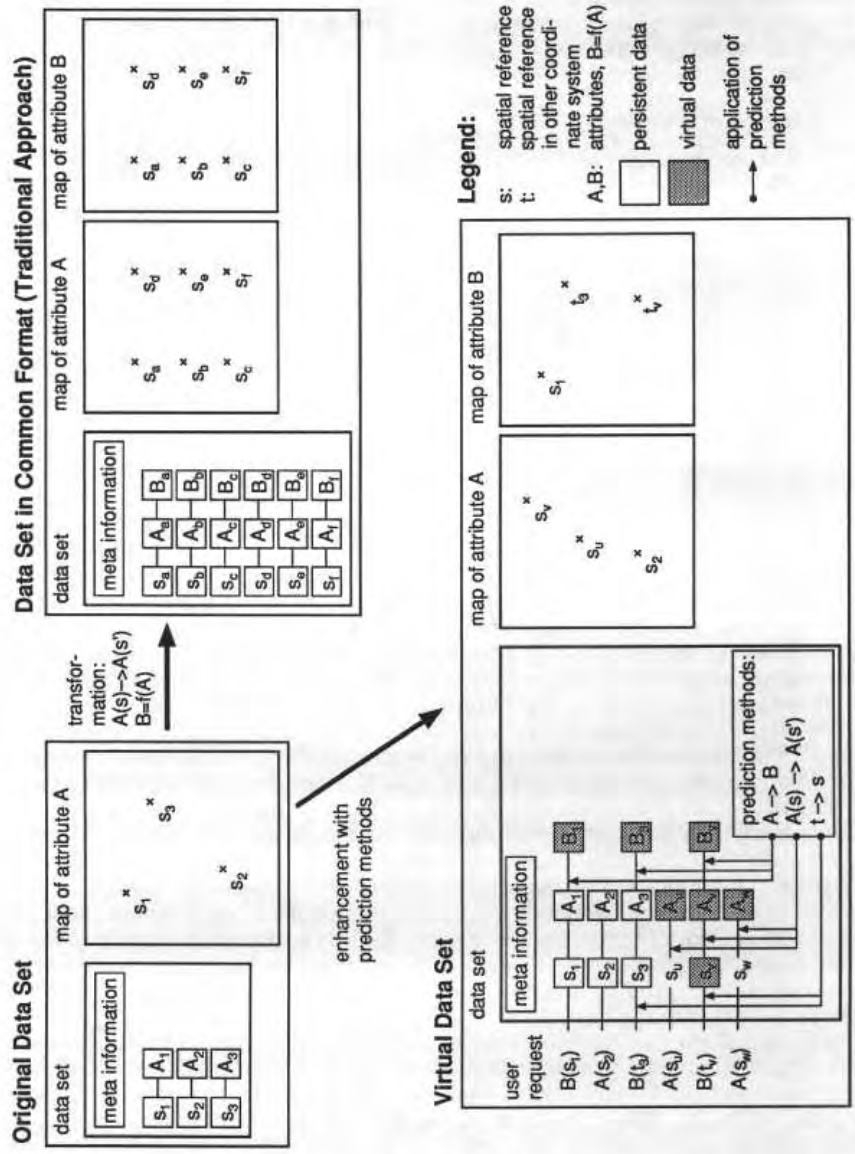


Figure 1: Static structure diagram of the homogenization of a data set using a common format and a virtual data set.

Comparing the virtual data set with traditional approaches one can see that it provides more flexibility, since there are no constraints by a given common format. Beyond that, a virtual data set is designed such, that data quality information is mandatory. The virtual data set enables the user to carry out any integrated analysis and see the effects of missing or unpredictable data as uncertainties visualized on the resultant maps; whereas traditional approaches limit the integrated analysis to data available in a common format, as long as the user does not perform additional transformations. The idea of a more flexible approach is also encouraged by the complexity of the operations involved particularly in environmental decision support. It is often very difficult or even impossible to estimate the influence of different input values to the result without the help of error models or sophisticated sensitivity analysis. Even very uncertain predictions may be of more value in an integrated analysis than no value at all.

The virtual data set also encapsulates the expert's knowledge (choice of the appropriate methods, meta information) within the data set, which will presumably lead to better predictions and reliable quality informations.

The virtual data set is able to provide values at very high resolutions when requested. So it is important to avoid the common misunderstanding that data of high resolution are implicitly more accurate than data of coarse resolution.

Figure 2 compares the data integration process with respect to the data flows in the traditional approach and the virtual data set for a sample integrated analysis.

SOME REQUIREMENTS FOR USING VIRTUAL DATA SETS IN GIS

Having introduced the general idea and some theoretical background of data integration on the basis of virtual data sets, we will now concentrate on some implementation issues. Commercial GIS do not meet the requirements for handling virtual data sets. An essential need is comprehensive data quality handling including error propagation and data quality visualization. Furthermore, the use of an object model will simplify the implementation of the virtual data set concept.

Object Model and Persistence

One of the key ideas of the virtual data set is the need to have the prediction methods stored with the original data. This, together with the needs for high level information hiding or encapsulation, suggests the application of the object model and object-oriented design (Booch, 1991) for describing the structure of the virtual data set.

A general problem is the required persistence (i.e. storage) of the prediction methods. One of the motivations of the virtual data set was the encapsulation of expert knowledge (e.g. prediction methods) within the data set. This will enable an application-independent use of the data. Data and procedures must thus be included in a data set. While current data base management systems (DBMS) offer little support for procedural data, especially when they have to be stored together with the 'normal' data within the data base, there are some approaches to enhance an (object-oriented) DBMS to allow storage of procedural data (e.g. Deux, 1991; Heuer, 1992). It is, however, difficult to establish similar capabilities for persistence and transfer between systems for procedural data since procedural data often depend heavily on characteristics of processors, compilers or interpreters. Some promising work adressing the integration of distributed systems with an *object request broker architecture* is presented by the Object Management Group (Soley, 1992).

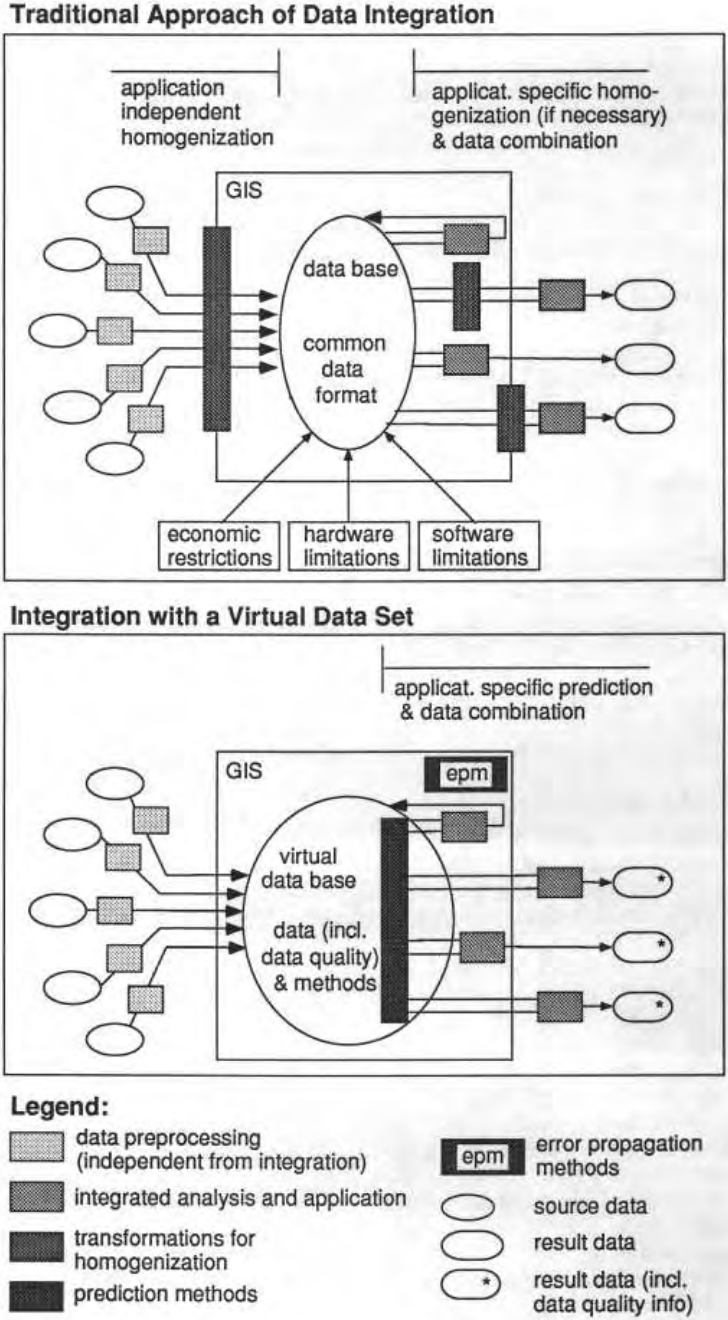


Figure 2: Comparison of data integration process in traditional approach (upper) and virtual data set (lower).

Error and Uncertainty Propagation

In the previous sections we always assumed that GIS are capable of handling data quality information transparently. This assumption is not very realistic. The heterogeneity of the data sets and the complexity of the operations performed demand more efficient and highly integrated capabilities for assessing uncertainty.

The concept of virtual data sets increases the importance of uncertainty handling by adding new sources of uncertainty. While many operations are forbidden in a traditional integration approach because of non-existent data, the virtual model allows virtually any operation between two or more data sets. The difference is, that the virtual data set might deliver absolutely uncertain values at locations or points in time when there is no reliable prediction possible.

Currently, considerable research efforts are on the way with respect to data quality and error propagation models in operational GIS. For example, Wesseling and Heuvelink (1991; 1993) present a system based on second order Taylor series and Monte Carlo methods among others to estimate error as a result of operations on stochastic independent and dependent uncertain input values.

In addition to those ideas we suggest the use of interval mathematics (e.g. Moore, 1979) for an easy to implement and very conservative error propagation scheme (i.e. error is never under-estimated, but often over-estimated). Especially for environmental applications the complexity of the problems encourages the use of conservative error estimates.

Interactive Visual Support for Data Integration

Interactive visual support is an important component of data integration. In order to be able to use GIS as a decision support system, it should be highly interactive and present graphical results. On the one hand the system needs to acquire expert knowledge in a communicative and exploratory way, such as the decisions leading to the selection of appropriate prediction methods to create a virtual data set.

On the other hand the system should be capable of visualizing data quality. While the need is clear from the above considerations, the solution is not. Data quality information adds several new layers of information in a GIS data base. Its visualization needs, therefore, to be able to handle multidimensional data.

Verification and validation of the selected or newly defined prediction methods should be supported in a standardized way. The subsequent users of the (virtual) data set will usually not reflect on the prediction methods defined and are therefore dependent on a consistent quality of the prediction methods[1].

CONCLUSIONS AND FUTURE RESEARCH

We have shown that major problems of data integration are heterogeneity and the lack of comprehensive data quality handling. The homogenization of data sets always involves prediction and thus adds further uncertainty. A flexible homogenization scheme is established with the help of the presented virtual data set. Enhancing it with the appropriate data quality models will facilitate an unrestricted, yet reliable integrated analysis.

Future research will concentrate on refinement of this concept and its realization. This demands a detailed application of the object model and the embedding of prediction methods and error propagation models.

---

1. It is very important to note that the *consistent quality* does not stand for the quality of the predicted virtual data. Rather, it guarantees that the prediction *method* meets a certain minimum quality requirement. This provides a kind of quality assurance during the process of the defining the virtual data set.

ACKNOWLEDGEMENTS

This work has been funded by the Swiss National Science Foundation under contract No. 50-35036.92. We would also like to acknowledge the contributions of Prof. Dr. Kurt Brassel and Dr. Robert Weibel.

REFERENCES

Booch, G. 1991, *Object oriented design with applications*, Benjamin/Cummings Publishing Company, Redwood City

Burrough, P.A. 1991, The Development of Intelligent Geographical Information Systems: *Proceedings of the EGIS'91 Conference*, Brussels (Belgium), pp. 165-174.

Chrisman, N.R. 1989, Modeling error in overlaid categorical maps: Goodchild, M.F. and Gopal, S., *Accuracy of spatial databases*, Taylor & Francis, London, pp. 21-34.

Dangermond, J. 1989, The organizational impact of GIS technology: *ARC News*

Deux, O. 1991, The O_2 system: *Communications of the ACM*, Vol. 34, pp. 34-48.

Heuer, A. 1992, *Objektorientierte Datenbanken: Konzepte, Modelle, Systeme*, Addison-Wesley, München

Lodwick, W.A. 1989, Developing confidence limits on errors of suitability analyses in geographical information systems: Goodchild, M.F. and Gopal, S., *Accuracy of spatial databases*, Taylor & Francis, London, pp. 69-78.

Maffini, G., Arno, M. and Bitterlich, W. 1989, Observations and comments on the generation and treatment of error in digital GIS data: Goodchild, M.F. and Gopal, S., *Accuracy of spatial databases*, Taylor & Francis, London, pp. 55-68.

Maguire, D.J. 1991, An overview and definition of GIS: Maguire, D.J., Goodchild, Michael F. and Rhind, David W., *Geographical Information Systems: principles and applications*, Longman, London, Vol. 1, pp. 9-20.

Moore, R.E. 1979, Methods and applications of interval analysis: Society for industrial and applied mathematics, *Studies in applied mathematics*, Vol. 2

Nyerges, T.L. 1989, Schema integration analysis for the development of GIS databases: *Int. Journal of Geographical Information Systems*, Vol. 3, pp. 153-183.

Openshaw, S. 1989, Learning to live with errors in spatial databases: Goodchild, M.F. and Gopal, S., *Accuracy of spatial databases*, Taylor & Francis, London, pp. 263-276.

Piwowar, J.M., Le Drew, E.F. and Dudycha, D.J. 1990, Integration of spatial data in vector and raster formats in a geographic information system environment: *Int. Journal of Geographical Information Systems*, Vol. 4, pp. 429-444.

Rhind, D.W., Green, N.P., Mounsey, H.M. and Wiggins, J.S. 1984, The integration of geographical data: *Proceedings of the Austra Carto Perth Conference*, Perth, pp. 273-293.

Shepherd, I.D.H. 1991, Information integration and GIS: Maguire, D.J., Goodchild, Michael F. and Rhind, David W., *Geographical Information Systems: principles and applications*, Longman, London, Vol. 1, pp. 337-360.

Soley, R.M. 1992, Using object technology to integrate distributed applications: *Proceedings of the First Intern. Conference on Enterprise Integration Modelling*, Hilton Head, SC (USA), pp. 445-454.

Veregin, H. 1989, Error modeling for the map overlay operation: Goodchild, M.F. and Gopal, S., *Accuracy of spatial databases*, Taylor & Francis, London, pp. 3-18.

Wesseling, C.G. and Heuvelink, G.B.M. 1991, Semi-automatic evaluation of error propagation in GIS operations: *Proceedings of the EGIS'91 Conference*, Brussels (Belgium), pp. 1228-1237.

Wesseling, C.G. and Heuvelink, G.B.M. 1993, Manipulating quantitative attribute accuracy in vector GIS: *Proceedings of the EGIS'93 Conference*, Genoa (Italy), pp. 675-684.

An Implementation Approach for Object-oriented Topographic Databases using Standard Tools

Babak Ameri Shahrabi[1], Wolfgang Kainz

Department of Geoinformatics
International Institute for Aerospace Survey and Earth Sciences ITC
P.O. Box 6, 7500 AA Enschede, The Netherlands
phone: +31 53 874-434, fax: +31 53 874-335
e-mail: ameri@itc.nl & kainz@itc.nl

Abstract

The creation of (national) topographic databases is one of the high priority tasks of mapping agencies. There are already numerous projects in many countries such as the United States of America, France and Germany. They all use different approaches, but mainly based on conventional data models and database management systems.

This paper focuses on the design of an object oriented data model for large topographic databases and its implementation by using conventional database and GIS technology. The proposed approach is tested in the context of the national topographic database in the scale of 1 : 25000 of Iran. It is part of a larger project looking into alternative ways of implementing the national topographic database by using different approaches for data modelling and database querying.

Introduction

Manually drafted maps were the primary media to represent the location and relationships of geographic objects. As the user's knowledge of map capabilities increases analog maps proved to be insufficient for fast and effective extraction of information for geographic knowledge. Consequently, it became necessary to convert the knowledge stored on paper maps to digital form.

Within the realm of geographic data handling, this trend (increasing reliance on the computer as a data handling and data analysis tool) has been driven by both push and pull factors. The primary one was the push away from the limitation of manual techniques and the pull toward the use of computers (Peuquet and Marble 1990). The substantial improvement in computer systems during the last two decades has made it much easier to apply computer technology to the problem of storing, manipulating and analyzing large volumes of spatial data. Today many national survey organization make routine use of what are called topographic databases to undertake tasks such as production of base map series, more efficient and cost effective map updating, and support several user needs which deal with the analysis of spatial information. In general terms, a topographic database contains an image or model of real world phenomena. This model is an organized collection of related data about geographical

[1]Permanent address: National Cartographic Center of Iran (NCC), P.O.Box 13185-1684, Tehran, Iran. Fax: +98 21 6001971

features which can be divided in two types, spatial (geometry) and non-spatial (alphanumeric attributes) data. Accordingly, two types of database models should be distinguished: hybrid and integrated models (Healey 1991, Wessels 1993).

Hybrid systems store spatial data in a set of independently defined operating system files for direct, high speed access. Each file (also called maplayer or maptile) represents a selected set of closely associated topographic features (e.g. river, railway, boundary). Non-spatial data are stored in a standard database, usually a relational one which is linked to the spatial data through identifiers.

Integrated systems store both the spatial and non-spatial data in a (relational) database. These systems do not define independent layers or tiles, but just one seamless map. Relational tables hold the coordinates, topological information and attributes.

The efficient management of very large geographic and topographic data sets is a highly complex task that current commercial systems have great difficulties with. Conventional DBMSs usually support data models that are not very well suited for the management of geographic information. Geographic objects are often structured in a hierarchical manner. For example a province consists of several cities which are split up into districts, which in turn consist of streets, blocks, etc. Neither conventional DBMS can provide sufficient support for these complex geo-objects. Instead the user typically has to split up objects into components until they are atomic. This fragmentation considerably complicates the modeling which consequently complicates the interpretation of modeled objects and may have negative effects upon the performance of the system.

Recent research in software engineering has promoted an object-oriented design method by which real world objects and their relevant operations are modeled in a program which is more flexible and better suited to describe the complex real world situation. Object-orientation is a particular view of the world which attempts to model reality as closely as possible to applications (Webster 1990, Wessels 1993). The notion of an object made its first appearance in the early 1970s as the part of the simulation language SIMULA. But, since its influence is detectable in several distinct branches, it is also true that developments within those disciplines have helped to refine the set of unifying concepts that define the object-oriented approach and it should no longer be seen solely as a programming style (Webster 1990).

Object-orientation is applied to information technology in several ways (Kim and Lochovsky 1989, Bhalla 1991). Possible applications, relevant for spatial information systems are object-oriented programming languages, object-oriented databases and user interfaces (Worboys et al. 1990, Kemppainen and Sarjakoski 1990, Khoshafian and Abnous 1990).

Tools

At the present time most geographic information system designers apply relational database technology to implement their model for two important reasons. Firstly, relational database management systems are dominant on the database market and enjoy wide acceptance among database users; they are a 'de facto' standard for data processing applications (Wessels 1993). Secondly, internationally standardized tools,

such as SQL, have been established for relational DBMSs which provide features for defining the structue of the data, modifying the data, and specifying security constraints within the database. Users can retrieve data by specifying in SQL what data are required. The system will work out the way how the requested data is retrieved (Date 1986).

However, object-oriented DBMSs are a new database technology and are therefore not much present on the database market. The fact that standards, as those for relational DBMSs are still not defined, and also as most of the organizations already have an information system based on a relational DBMS technology, the former approach is still the most important technology applied to spatial information systems. Whilst there is some general agreement on the characteristics of object-oriented databases, there is no consensus on how an object-oriented database should be implemented (Worboys et al. 1990). At present there are no full-scale implementations, but a number of research efforts are in progress of creating object-oriented databases which can be classified into three categories (Bhalla 1991):

(1) Those that are directly based on the object-oriented paradigm like ONTOS (Ontos 1991),
(2) extensions to relational systems such as POSTGRES (Stonebraker et al. 1990), and
(3) experimental toolkits (or storage system) such as EXODUS (Carey et al. 1986).

The second category indicates that relational DBMSs have developed a variety of techniques to facilitate object management within a database environment. These include storage of the structure of object hierarchies in relational tables, inheritance of instance variables and methods, storage of query language or programming language procedures representing instance methods, as specified fields in relational tables, and support for abstract user defined data types and operations supported on them. It should be considered that this approach is an evolutionary method as the best way forward to add object-oriented facilities to an existing relational database framework. This is in contrast to some of the proponents of object-oriented programming, who wish to achieve the same kind of functionality, but without the perceived constraints of the relational model (Healey 1991).

Query languages are very important for the retrieval of data to satisfy certain immediate user needs which were not planned for and for which no programmed access procedures are available. Interactive languages for ad hoc queries are usually included in conventional DBMSs. The SQL (Structured Query Language) whose original version was known as SEQUEL is the language which rapidly emerged as the standard database language and is widely implemented. SQL allows users to work with higher level data structures. Rather than manipulating single rows, a set of rows can be managed. SQL commands accept sets or rows as input and return sets as output. This set property of SQL allows the results of one SQL statement to be used as input to another. SQL does not require users to specify the access methods to the data. This feature makes it easier to concentrate on obtaining the desired results. A number of attempts have been made to extend SQL to make it useful for spatial databases (Egenhofer and Frank 1989, Egenhofer 1991). The PL/SQL, a procedural language extension to SQL (ORACLE 1992), fully integrates modern software engineering features which are practical tools for system developers and designers to bridge the gap between conventional database technology and procedural programming languages.

Another aspect is the programmers interface or procedural programming available in most GISs to provide a tool for developing user-specific interfaces such as AML (Arc Macro Language of Arc/Info), or a graphical user interfaces such as ESRI's ArcView.

Implementation

The object-oriented topographic data model is based on the object concept. The model represents a real world phenomenon of whatever complexity and structure as an object (Dittrich and Dayal 1986, Bonfatti et al. 1993). An object is characterized by encapsulating its structural/static properties (attributes, relationships, components) and its behavioral/dynamic properties (rules, operators, procedures) (Egenhofer and Frank 1988, Webster 1990).

Data Modelling

The components of design of an object-oriented model are as follows :

I- Object identification
Objects are the individual elements in the model which are defined for applications. In this research the objects are based on the specification and the contents of the base map series 1:25000 of Iran as a minimum requirement. However, they are modified in such a way to satisfy as much as possible other user groups.

II- Classification
Classification is the mapping of several objects into a common class. All objects that belong to the same class are described by the same properties and undergo the same operations. For example, in the transportation network, all highways which have the same properties (structural and behavioral) can be classified as a class *highway*.

III- Generalization and Inheritance
Similar to the concepts of object and class, the object-oriented model also supplies the dual concepts of generalization and specialization. Together, these two allow classes to be arranged in a hierarchy. A class which is a descendant of another class in the hierarchy is said to be a more specialized subclass, while the more general class is the superclass. A subclass has automatically the characteristics of the superclass defined for it but it may have also specialised characteristics of its own. Subclass and superclass are related by an IS_A relationship. For example, the classes *highway*, *main road*, *secondary road* could be grouped as a superclass *road*. In generalization hierarchies, some of the properties and operations of the subclasses depend on the structures and properties of the superclass(es). Properties and operations which are common for superclass and subclasses are defined only once, and the properties and operations of the superclass are inherited by all the objects of the corresponding subclasses. This concept is very powerful, because it reduces redundancy and maintains integrity. The inheritance can be single or multiple. Single inheritance requires that each class has only one direct superclass. In case of multiple inheritance, one subclass has more than a single direct superclass. For example, an object channel could be a subclass of two different superclasses, such as *water stream*, and *structure*. It will inherit properties from both superclasses. Sometimes the same property operation can be defined differently in two relevant superclasses, which is known as an inherit conflict. For

solving these conflicts, some special conditions must hold or a priority rule must be defined.

IV- Aggregation
In addition to elementary objects composite objects may exist. They can be defined through aggregation. An aggregation shows how composite objects can be built from elementary objects and how these composite objects can be put together to built more complex objects and so on. Suppose that we have houses, roads, parks as simple objects, then from these we can build complex object residential districts. The fact that the simple objects can be aggregated into complex objects implies that also their attribute values may be aggregated. If one of the attributes of object house is the number of people living there, then it is easy to calculate the total number of inhabitants of a residential district.

V- Association
Association or grouping is the construct which enables a set of objects of the same type to form an object of higher type. Association is used to describe the logical relationship between objects (classes). The association can be established between two or more classes or recursively in one class. For example, the association between road and railway is an association which describes that, road and railway are crossing each other. A road might cross the same railway several times. Both road and railway are members of the relevant association.

The features of interest in the topographic model include 9 classes at the top level. These spatial information classes are :

Survey control point consists of information about the points of established position that are used as fixed references in positioning the features. This class of objects includes three subclasses such as *geodetic control point* (planimetry), *bench marks* (altimetry), and the object class *full control point* (both planimetry and altimetry).

Vegetation consists of information about vegetative surface cover. It will be specialized to classes such as *forest, orchard, arable land* and so on. Vegetative features associated with wetland such as marsh and swamps are collected under class *hydrography*.

Building consists of information about the object building which means any permanent walled and roofed construction. This is a class including several subclasses such as class *educational building, medical building* etc. that each one is again a generalization of several classes. For example the object class *hospital*, or *clinic* are subclasses of class *medical building*.

Hydrography describes all information about the hydrographic features and includes *water course, water bodies*, and *water surface points* as its subclasses. These classes again are specialized to several classes. For example the *water course* is a superclass of object class *river, canal* etc.

Delimiter consists of all boundaries and delimiters including virtual and actual boundaries such as delimiters of a park or a city.

Road & Railway describes all spatial features which provide a track for moving cars, locomotives, humans etc. from an origin to a destination.

Utility consists of all the information about the public services such as gas, water, electricity, etc.

Structure consists of all the structured spatial data which means all the construction on a given site which is not roofed and walled, such as a bridge, dam, airport runway, etc.

Hypsography includes all the information about the land form and relief.

Properties and relationships among spatial objects

The topographic model is structured according to the object-oriented concept. This concept describes different relationships and hierarchies among the classes and objects. Figure 1 presents part of the proposed model with the relationships among the object classes. The following is the description of the model corresponding to Figure 1. The role of this example is not to present a complete definition but only to illustrate the approach.

class *Road & Railway*

Attributes ;
OID : integer
Name : string
Length : real
Date-of-construction : date
Pass-above : list of bridges
Pass-under : list of bridges
Pass-through : list of tunnels
Intersection-road : list of roads
Intersection-railway : list of railways
Last-maintenance : date

Constraints ;
Length > 0
last-maintenance >= date-of-construction

Class methods ;
Create
Delete
Modify

Object methods ;
Draw
.
.

The class *Road & Railway* has a relationship Pass-under and Pass-above with the object class *Bridge* which means a road or a railway may pass the roads, railways, or rivers via the bridges. It also has a relationship Pass-through with the object class *Tunnel* which means a particular road or railway should be passed through tunnels.

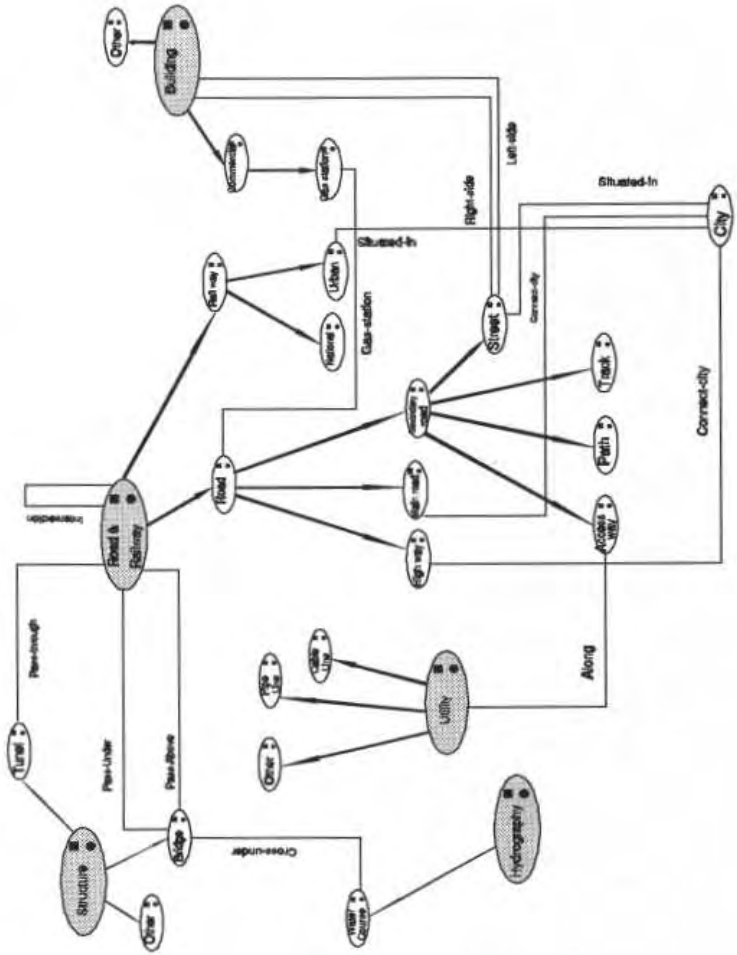


Figure 1: *Relationships among the object classes*

Class *Road* also has its own properties as follows :

class *Road* is a subclass of *Rail & Railway*

Attributes ;

Width : real
Surface : string
Lanes : integer
Gas-station : list of gas stations

Constraints ;

Width > 0

Class methods ;
Object methods ;

There is a relation Gas-station among the class *Road* and the class *Gas station building* which indicates all available gas stations along a particular road.

class *Secondary road* is a subclass of *Road*

Attributes ;
Direction : integer

Constraints ;
Direction = 1 or 2

Class methods ;
Object methods ;

class *Street* is a subclass of *Secondary road*

Attributes ;
Situated-in : city
Right-Side : list of buildings
Left-side : list of buildings
Traffic-flow : real

Constraints ;
Class methods ;
Object methods ;

In this research only some parts of the proposed model including the most important aspects of the object-oriented approach are considered and implemented. The practical work is done using Oracle and Arc/Info. PL/SQL and AML are used for the implementation of data types and methods.

Conclusions

Object-orientation is currently the most promising approach to spatial data handling. Among the several ways of introducing these concepts in GIS is implementing object-oriented features by using conventional tools. This can be done for instance by taking existing (conventional) software systems and "emulating" object-orientation by putting an object shell on top of them using available data definition and (macro) language facilities. This research must be seen in the context of a larger research program on intelligent geographical information systems. One of its goals is to investigate alternative ways of implementing topographic databases by using conventional (available) software systems.

References

Bhalla, N. (1991): Object-oriented data models: A perspective and comparative review. In: Journal of information science (17), pp. 145-160.

Bonfatti, F., Cantaroni, R., Gentili, L., Murari, C. (1993): Object-oriented support to the design of Geographic information systems. In: Proceeding of EGIS., Vol. 1, pp. 754-763.

Carey, M.J., Dewitt, D.J., Richardson, J.E., Shekita, E.J. (1986): Object and file management in the EXODUS extensible database system. In: Proceeding of 12th international conference on VLDB, Kyoto, Japan, pp. 91-100.

Date, C.J. (1986): An Introduction to Database Systems. Volume I, Reading, Massachusetts, Addison-Wesley Publishing Company.

Dittrich, K., Dayal, U. (Ed) (1986): Proceedings of the international workshop on object-oriented database systems., IEEE computer society press, Asilomar, California, New york, USA.

Egenhofer, M.J. (1991): Extending SQL for graphical display. In: Geography and Geographic information systems, Vol. 18, No. 4, pp. 230-245.

Egenhofer, M.J., Frank, A.U. (1988): Object-oriented modelling: A powerful tool for GIS. Seminar workbook 'object-oriented technology' presented november 29/1988, San Antonio, Texas, Published by: NCGIA.

Egenhofer, M.J., Frank, A.U. (1989): Object-oriented modeling in GIS: inheritance and propagation. In: E. Anderson (Ed), Proceedings of Auto Carto 9, American Congress on Surveying and Mapping, American Society for Photogrammetry and Remote Sensing, Falls Church, pp. 588-598.

Healey, R.G. (1991): Database management systems. In: Geographic information systems. Volume I, principles, edited by: Maguire, Goodchild, Rhind, Longman, Scientific & technical, Bath press: Avon

Kemppainen, H., Sarjakoski, T. (1990): Object-oriented GIS: A review and a case study. In: Proceeding of ISPRS commission III, China.

Khoshafian, S., Abnous, R. (1990): Object Orientation, Concepts, Languages, Databases, User Interfaces. John Wiley & Sons.

Kim, W., Lochovsky, F.H. (1989): Object-oriented concepts, databases and applications. ASCM press, New York.

ONTOS (1991): ONTOS Developers Guide. Ontos Inc., Burlington Massachusetts, Three Burlington Woods.

ORACLE (1992): PL/SQL user,s guide and references. ORACLE manual, ver. 2.0.

Peuquet, D.J., Marble, D.F. (1990): Introductory readings in geographic information

systems. Taylor & Francis.

Stonebraker, M., Rowe, L.A., Hirohama, M. (1990): The implementation of POSTGRESS. In: IEEE Transaction on knowledge and data engineering, 2 (1),pp. 125-142.

Webster, C. (1990): The Object-Oriented paradigm in GIS. In: Proceeding of ISPRS, Commission III, China.

Wessels, C. (1993): Object-orientation and GIS. In: Proceeding of EGIS., Vol. 2, pp. 1462-1471.

Worboys, M.F., Hearnshaw, H.M., Maguire, D.J. (1990): Object-oriented data modelling for spatial databases. Int. J. Geographical Information Systems, Vol. 4, No. 4, pp. 369-383.

CONVEYING OBJECT-BASED META-INFORMATION

by

Peter F. Fisher

Midlands Regional Research Laboratory
Department of Geography, University of Leicester
Leicester LE1 7RH, United Kingdom

pff1 @ leicester.ac.uk

ABSTRACT

Metadata and lineage are two related areas of research which have received considerable recent attention from the GIS community, but that attention has largely focused on meta-information at the level of the image, coverage, or data layer. Researchers working in other areas such as error and uncertainty handling, have focused on a lower level within the spatial data, but can also be considered to have been working on metadata. Users require access to the whole set of metadata from the level of the mapset to the elemental object, i.e. the point, line, polygon or pixel. This paper attempts to draw these different research strands together and suggests an umbrella framework for metadata which can be translated to a number of different flavors of metadata for which accuracy, lineage, statistics and visualization are exemplified. This leads to discussion of an interface design which enables rapid access to the metadata.

INTRODUCTION

Meta-Information or Meta-Data have been defined merely as data about data (Lillywhite, 1991). It is therefore rather surprising to find that two relatively restricted areas which fit this definition have been the focus of most research associated with metadata. One line of investigation has come from the database community and focuses on datasets as a whole; the properties and contents of each dataset being the prime information of interest (Medjyckyj-Scott et al. 1991). This research is epitomised by such systems as, for example, BIRON (Winstanley, 1991), FINDAR (Johnson et al. 1991), MARS (Robson and

Adlam, 1991) and, more recently, GENIE (Newman et al. 1992). On the other hand, lineage has been addressed by Lanter (1991) who developed the GeoLineus system to track coverage transformations in Arc/Info, an approach which is being extended to other systems. Combined with error reports it is also possible to use the lineage information to propagate overall errors (Lanter and Veregin, 1992). While these two lines of research have been identified as associated with metadata and dominate that literature, other research is being done, or the need for it is realised.

For example, under the NCGIA initiatives on the accuracy of spatial databases (I-1) and the visualization of spatial data quality (I-7) much work has been done on the error measurement, reporting, modeling, propagation, and visualization. This is all to do with metadata, since accuracy information is not usually the data, it is about the data. The most recent initiative is on the Formalization of Cartographic Knowledge (I-8), and is among many other Artificial Intelligence developments in GIS. The result of reasoning from an AI system is new data, but the process of reasoning is metadata.

The purpose of this paper is twofold. First is to suggest an ***umbrella*** framework for spatial metadata within GIS which encompasses all the strands of research outlined above as well as others which are not mentioned and which may yet emerge. The second theme to the paper, is to examine how these may be offered to a user for exploration and interrogation, both as cartographic and as textual presentations.

A HIERARCHY OF OBJECTS IN A SPATIAL DATABASE

Moving from the most general to the most specific, there is a logical hierarchy of objects in the spatial database. This hierarchy is recognised in the design of many systems, but it is necessary to recall it for the present discussion. The hierarchy is listed in Tables 1 and 2 where it is given in order from the most general to the most specific.

The Mapset is a collection of data about a particular area. Usually all members of the mapset are to a common projection (itself one of the metadata items of the mapset), and covering an identical area (although this is not essential). It is the highest level of metadata which includes spatial and aspatial information. One essential item of metadata at this level is the contents of the next level in the hierarchy, the data layers.

The Data Layers (based on a layer-based system but also transferrable to an object-oriented one) are the individual themes of

information. They are variously referred to as the image, the raster, the quad, the coverage, etc. in different systems. Arguably in a true object-based system the Mapset and the Data Layer are one and the same level, but in most current systems they are distinct.

The Object Classes is the individual theme on the thematic map. So in a soil Data Layer, the Object Classes would be the particular soil map units, while in a census area data layer, the object class would be the Enumeration District, or Census Block. In a topogaphic layer it would include all the features types present.

The Spatial Objects are the elemental spatial features which GIS are designed to work with. These may be points, lines, areas or pixels. Surface data being usually coded as lines, points or pixels. Commonly one derivative data layer can be the metadata for a source layer.

TABLE 1

RAW OBJECTS

Object-Types:	**Examples and Some Key Characteristics**
The Mapset	The overall group of data - the study area
The Data Layer	A theme over the mapset area - Elevation - Soil data - Census areas
The Object Class	Individual map themes - Soil type X - Wetlands - Public Buildings - Each Block
The Spatial Object	Objects actually displayed - Point - Line - Polygon - Pixel

The four levels recognised here are identifiable in a number of different versions.

METADATA EXAMPLES

Some types of metadata can be tracked throughout the hierarchy identified above, and the examples of *accuracy*, *lineage*, *statistics* and *visualization* are discussed.

Accuracy

Accuracy can be tracked from the highest level to the lowest. At the *mapset* level it refers to the accuracy of the global coordinates to

which the mapset is registered, while at the *data layer* level, it refers to the accuracy of any spatial transformations the layers may have undergone to become registered to the mapset area and projection; both are measures of positional accuracy. Occasionally all data layers will have the same error reports and so it may be recorded at the mapset level. At the data layer level attribute accuracy may also be recorded as the Percentage Correctly Classified, the Overall Accuracy, or the Root Mean Squared Error. These are all different broad measures of accuracy used for different data types including soils, remote sensing/land cover and elevation.

TABLE 2

ACCURACY METADATA

Object-Types:	Examples and Some Key Characteristics
The Mapset	- Coordinates of control points on some wider-area reference system (long/lat)
The Data Layer	- Percent Correctly Classified (overall accuracy) - RMS Error
The Object Class	- Producer and User accuracy - Map unit inclusions
The Spatial Object	- accuracy of classification - positional accuracy - accuracy of area estimates

The *object class* may be associated with a number of different types of accuracy measures. In soil data this may include the types and frequency of map-unit inclusions (Fisher, 1991), and in land cover data it may include the producer and user accuracy (Campbell, 1989). Among the *spatial objects* in any polygon coverage there will be a unique value for the accuracy of the area of the polygon, while lines may have a unique value for the accuracy of the length of the line. In a classified remotely sensed image the fuzzy memberships of each and every pixel belonging to all land cover types may be recorded, although these have to be considered as a data layer in their own rights as well as metadata to the usual hard classification. Accuracy in the *viewed objects* is very poorly researched, but may include a degree to which human observers may confuse the symbolism used for one object class with that used for another object class in the display space. Metadata on the viewed spatial objects may report the degree to which the object has been deformed in the process of cartographic representation (the number of points removed, the ratio of the length shown and the actual length, etc.)

Lineage

Spatial datasets are derived by some process, either by analysis of other data layers or by digitizing. The history of these transactions and events is the lineage of the data. At the level of the *data layer*, Lanter (1991) has given a thorough treatment of lineae issues, particularly developing Geolineus, a system which uses history files to produce the events in the lineage of any *data layer*, and incorporating some further measures such as error propagation (Lanter and Veregin, 1992). Various widely available GIS also give some measure of lineage of *spatial objects*, or can be forced to. EPPL7 and GRASS, for instance, enable point-and-click presentation of spatial objects (pixels) in multiple data layers, although the default method of presentation in both is merely to display one layer and list the values in other, named layers, which may be related by lineage or just by being of the same area. The user needs to know the lineage of any data layer for this method to work. Lanter and Essinger (1991) illustrate a more automated version of this, but again no impression of lineage is given.

As in accuracy, lineage is a multilevel concept within the hierarchy. The *mapset's* lineage may include the subsetting of the basic layers from some other dataset at some other projection (in a very diverse *mapset*, this information may be specific to the *data layer*). Data layers need to record how they are individually derived, and so for data layers at the root of a lineage tree, this may include the name of the digitizer, and the scale of the paper map. In derived data layers they it records progressively the parent data layers.

By cross referencing to the parent data layers and the conditions used to execute the transformations, it is possible to present the user with lineage information on the *object classes*, and *spatial objects*. With object classes the lineage data should report the threshold values which transformed other classes in the derivation of the current class, while for spatial objects the user should be presented with parent values at that location.

The lineage of the *viewed object* identifies any transformations made to either object classes or spatial objects which have been executed for the purpose of display only; the change from a numeric value to a color on a screen (in some systems this is a hard numerical transformation, in others it is programmed, but in still others it may be specified).

Lineage as the term is used here refers to logical or mathematical transformations of data. It is therefore not dissimilar to logical arguments, and the logical inference of an expert system may also be

viewed in the same way. Indeed Lanter's (1991) flow diagram presentation of lineage (see also Berry, 1987), is very similar to the presentation of goals and rules in a graphic representation of an expert system's logic, and is one of the best ways to show a user the reasons for an expert system's judgement.

Statistics

Statistical reports should perhaps be regarded in the same general framework as the examples of metadata discussed above; going back to the definition of metadata statistical summaries are information about the data. Indeed, many of the accuracy measures widely recognised as metadata are statistical summaries of the hierarchical level. A consideration of the hierarchies presented should lead to an appreciation that there are many other summary statistics which can be presented at the different levels identified. For example, the correlations and regressions between different data layers is metadata at the level of the *mapset*, the spatial autocorrelation is at the level of the *data layer*, the *object class* or even the *spatial object* (depending on the measure used).

TABLE 3

STATISTICAL METADATA

Object-Types:	Example Visualization Metadata Types
The Mapset	- Inter-layer correlation matrix
The Data Layer	- Spatial autocorrelation
The Object Class	- Join counts statistic
The Spatial Object	- local correlation

Furthermore, and less commonly recognised, such summary statistics all have equivalents in the *viewed objects*. The measures may be similar but the values will frequently be very different. Arguably one specific objective in cartographic reproduction is to match such summary statistics between the viewed objects and the raw data (Dykes, in press). Thus a very important item of metadata can be the degree of match between the values in the raw data and in the viewed objects.

Visualization

The visualization of the raw data has a hierarchical set of unique metadata, whether on a screen or on paper. These are exemplified in Table 4. At the level of the Mapset there are properties of the projection used for display, and the characteristic pattern of distortion. It may need to be recalled that the display projection is not necessarily the same as the digitizing projection or the projection for storage, and

that final projection necessarily introduces its own artifacts into the data.

The name of the palette used to color a data layer, and its properties are crucial to how the user views the data; how they see it. In a situation where many windows may be in use at once, many different palettes may be in use. This is independent of the object class (legend) since that can be populated with any palette. The reasons for the use of the palette may be no more than it being the default, but they may be deeper.

The reasons for each object class being painted the color or using the symbol shown may be metadata at the next level. Some measure of the degree to which classes may be confused would also be useful information assisting design.

The mapped data groups items by class, so all rivers are shown as blue lines. The actual names of those rivers are then metadata, and a point and click system can give the user access to the source information. Where many different groups are classed as one for mapping, that too can be presented to the user as metadata. In another sense the algorithm used for generalizing some feature for display may be shown, and indeed literally the original detail of the feature is metadata to the visualization.

TABLE 4

VISUALIZATION METADATA

Object-Types:	**Example Visualization Metadata Types**
The Mapset	- The map projection used for display - Distortions caused by the projection
The Data Layer	- colour palette in use - reasons for using this palette - objectives of the design
The Object Class	- reasons for the color/symbol selection - confusion between classes
The Spatial Object	- names and actual classes of the features shown - the number of points omitted in generalizing a line

A USER INTERFACE

A graphic user interface is proposed to enable access to the different levels and types of metadata. Figure 1 shows the broad structure of such an interface in the display space there are a number of different live areas, each of which will produce a response from the system giving metadata on a particular object within any hierarchical class.

A menu of metadata types of which the system is aware needs to be shown. System awareness could be enabled by either documentation files, history files or history fields within documentation files, and the like. How this is enabled is not the subject of this paper. Selecting from this menu would cause changes in the actual metadata displayed. Within a full GIS, this could be a pull-down metadata menu or sub-menu from the main system menu, it would therefore be another aspect of the GIS functionality, presumably in the area of system control.

We then have a set of 4 active areas (possibly distinct windows) each showing an active area with or without subareas, and clicking within these areas will cause different levels of metadata to be displayed. Area 1 is the active area for the mapset, and Area 2 is for the data layer. The object classes corresponding to the data layer presented in Area 2 are then shown in Area 3, and the corresponding spatial objects are shown in Area 4. In these last two areas the individual themes and objects are themselves active and metadata on those can be retrieved by pointing and clicking. The metadata-type selected from the metadata menu is then displayed. A further menu may offer alternative methods for displaying the metadata. Thus error may be displayed in textual or many cartographic forms.

Multiple triplets of data-layer/object-class/spatial-object windows could exist within a mapset, and be toggled in a seventh window. As currently envisaged the display would only show one data layer at any time, but this is not a necessary restriction in a multiwindowing operating system.

CONCLUSION

In this brief paper it has been possible to present an argument which suggests that it is necessary to develop a complete view of metadata. Users wishing to find all the information about their data (at whatever level) can only be satisfied by such a complete view being taken. The nuts and bolts of this are essential, are being actively researched by many groups, and implemented in commercial systems.

A number of important issues do emerge from the discussion:

1. the umbrella structure for metadata is suggested as a necessary recognition;
2. the large number of possible different types of metadata;
3. the extent to which data, with its own metadata, are themselves metadata for derivative data layers must also be recognised;
4. the need for a consistent interface giving access to all this metadata should become an important part of any system.

Knowledgable users are aware of the number of different pieces of information which are available for data sets they are manipulating. They may often be frustrated by failure to access that information. Indeed, arguably all users deserve access to this information, since it is exactly this information which informs the user as to the suitability of the data and the analysis they have performed. But that access needs to be through a consistent and complete interface design itself based on a complete recognition of the metadata.

REFERENCES

Berry, J., 1981. Fundamental operations in computer-assisted map analysis. *International Journal of Geographical Information Systems*, Vol. 1, pp. 119-136.

Dykes, J., in press.

Johnson, D., Shelly, P., Taylor, M., and Callahan, S., 1991. The FINDAR Directory System: a meta-model for metadata. In Medjyckyj-Scott, D.J., Newman, I.A., Ruggles, C.L.N., and Walker, D.R.F. (editors), *Metadata in the Geosciences*, Group D Publications, Loughborough, pp.123-138.

Lanter, D.P., 1991, Design of a lineage-based meta-data base for GIS: *Cartography and Geographic Information Systems*, Vol. 18, pp. 255-261.

Lanter, D.P., and Essinger, R., 1991, *User-Centered Graphical User Interface Design for GIS*. Technical Paper 91-6, NCGIA, Santa Barbara.

Lanter, D.P., and Veregin, H., 1992, A research paradigm for propagating error in layer-based GIS: *Photogrammetric Engineering and Remote Sensing*, Vol. 58, pp. 825-833

Lanter, D.P., 1993, A lineage meta-database approach toward spatial analytic database optimization: *Cartography and Geographic Information Systems*, Vol. 20, pp. 112-121

Medjyckyj-Scott, D.J., Newman, I.A., Ruggles, C.L.N., and Walker, D.R.F. (editors), 1991, *Metadata in the Geosciences*. Group D Publications, Loughborough.

Newman, I.A., Walker, D.R.F., Mather, P., Ruggles, C.L.N., and

Medjyckyj-Scott, D.J., 1992. GENIE - The UK Global Environmental Network for Information Exchange, GENIE Research Report 1, Department of Computer Studies, Loughborough University.

Robson, P., and Adlam, K., 1991, A menu-aided retrieval system (MARS) for use with a relational database management system. In Medjyckyj-Scott, D.J., Newman, I.A., Ruggles, C.L.N., and Walker, D.R.F. (editors), *Metadata in the Geosciences*, Group D Publications, Loughborough, pp. 171-186.

Walker, D.R.F., Newman, I.A., Medjyckyj-Scott, D.J., and Ruggles, C.L.N., 1992, A system for identifying datasets for GIS users: *International Journal of Geographical Information Systems*, Vol. 6, pp. 511-527.

Winstanley, B., 1991, The approach of the ESRC Data Archive's BIRON system to the handling of spatial metadata. In Medjyckyj-Scott, D.J., Newman, I.A., Ruggles, C.L.N., and Walker, D.R.F. (editors), *Metadata in the Geosciences*, Group D Publications, Loughborough, pp. 115-122.

BIOGRAPHICAL SKETCH

Peter Fisher has degrees from Lancaster and Reading Universities and Kingston Polytechnic. He has taught at Kingston, at Kent State University, OH, and is now lecturer in GIS at Leicester University, where he directs the MSc in GIS and is an associated researcher with the Midlands Regional Research Laboratory. His recent research has been in the reporting, modelling, propagating and visualizing error and accuracy in spatial data. He is soon to be editor of the International Journal of Geographical Information Systems.

EMAPS: AN EXTENDABLE, OBJECT-ORIENTED GIS

Stephen M. Ervin, Associate Professor
Department of Landscape Architecture
Harvard University Graduate School of Design
48 Quincy St. Cambridge MA 02138 USA
(617) 495 2682 FAX (617) 495 5015
email: servin@gsd.harvard.edu

ABSTRACT

EMAPS is an interactive interpreted object-oriented environment for manipulating and displaying geographic information in raster (grid-cell-based) form. A hierarchy of objects -- windows, maps, cells -- takes advantage of the inheritance of properties in the Common LISP Object System to allow easy prototyping and exploration of variations. In this system, cell values need not be integers or real numbers -- they may be arbitrary functions, allowing each cell to operate as a cellular automaton. Writing procedures using rows, columns, cell-values and neighborhoods is a powerful way of expressing cartographic modelling operations, including map overlay, image processing, and hill-climbing algorithms; an object-based windowing system makes the graphical interface friendly and flexible. EMAPS is particularly valuable for modelling and exploring dynamic interactions in landscape ecology, and should be useful for teaching basic concepts of programming geographic information sytems.

INTRODUCTION

Most geographic information systems (GIS's) provide tools for the storage, manipulation and diplay of geographic information, whether in raster (grid-cell) or vector/polygon form. The better of them come with "languages" or macro-interpreters enabling encapsulation of procedures and supporting the process of 'cartographic modelling' [Tomlin] for performing analyses. Most GIS languages, however, are not envisioned as programming environments with the usual associated documentation, editing and debugging tools, or other facilities. Their macro languages are typically not constructed with great attention to their denotational semantics -- they are typically *ad hoc*, extended and revised from version to version -- and sometimes lack such basic constructs as conditional control, iteration, or scoping of variables. Thus cartographic models are limited to those (mostly linear) sequences which are supported by the macro language, combined with any 'black-box' pre-packaged analyses provided by the GIS itself. These are often unsatisfactory.

On the other hand, environmental designers -- landscape architects, urban designers, regional planners -- and other professionals -- geographers, ecologists, et al -- who are interested in modelling are likely to resist the prospect of having to learn 'to program' in order to pursue their work. In order to support inventive, unrestricted use of modelling using GIS technology, a middle-ground is desirable, which both provides the rigorous and supportive environment enjoyed by software engineers,

and yet is built around the basic contructs of geographic analysis - maps, regions, values, cartographic modelling primitives.

EMAPS is just such a system: an interactive interpreted environment for manipulating and displaying geographic information in raster (grid-cell-based) form that is based on the principles of object-oriented design, and uses Common LISP as its underlying macro language. The object-oriented design provides a clear conceptual model and provides all the usual advantages of modularity and extendability; the use of LISP provides semantic rigor and a comfortable, productive programming environment for undertaking explorations in geographic analysis and design.

OBJECT-ORIENTED APPROACH

In object-oriented programming, such as the Common LISP Object System (CLOS) [Steele, Keene], or SMALLTALK [Goldberg], two principal ideas underlie all programs: methods for 'encapsulation' of both structure and behavior into 'objects', which maybe organized hierarchically such that some kinds of objects are generic (classes) while others instantiate or specialize the genus (instances); and 'message passing' as the means of communication between objects, such that all objects have an effective 'communications interface' specifying the messages to which they respond, and how they respond. Both of these characteristics may be found in traditional programming languages to greater or lesser degree, and may be thought of as simply implemented by procedure-calls in such languages.

The idea of object-oriented GIS is simple and appropriate: maps are objects with behavior, and a GIS or a cartographic model is simply a collection of specified maps with specified relationships between them. Object-oriented GIS may as well be vector-based as raster-based, or indeed, hybrid of the two. The system described here is primarily raster-based. A hierarchy of objects in EMAPS -- windows, maps, cells -- takes advantage of the properties of object-oriented Common LISP to allow easy prototyping and exploration of variations. The list of objects in EMAPS starts out small:

A WINDOW is an object capable of performing computer graphics for display, with characteristics such as size, colors, bit-depth, etc.

A MAP is an object representing the spatial distribution of some attribute(s), that is capable of displaying itself in a WINDOW; each MAP has a list of characteristics such as name, thematic content, scope, scale and resolution, etc. Each MAP is also linked to a KEY.

A KEY is an object that basically implements a two-column table, providing a correlation between a value or range of values and a display attribute, such as a symbol, color, pattern, etc. The KEY is used when a MAP is displayed in a WINDOW.

A CELL is an object which represents a spatial extent on the ground and encodes an attribute or attrbutes for that extent. Each cell has a

'value', and one or more methods for displaying itself, and reporting, updating or changing its value.

In EMAPS, each MAP is composed of a number of CELLS, typically arranged in a rectangular two-dimensional array as in an ordinary raster GIS map. Each cell has one or more value(s), typically numeric, which are typically displayed by means of a color or pattern on CRT or hardcopy.

It's important to note, however, that in EMAPS, CELL values are in fact *procedures*, not simply values. In its general form, a CELL is an object which can respond to the request 'What is your value?' by invoking its associated procedure, perhaps with appropriate arguments (and depending on the time and form of the request the answer may even vary.) In this way, each cell is in fact a 'cellular automoton': an object with a state which may change as some function of a number of inputs, typically derived from other automota's states. Cellular automata provide a reasonably well-understood general model [Toffoli & Margoulis] with considerable potential for managing and exploring geographic information and environmental phenomena [Itami, Hogeweg].

EXTENDABLE CARTOGRAPHIC MODELLING

EMAPS is well suited to developing models involving geographic data, since the cellular automaton model embedded in object-oriented maps provides for all the inter-map (overlaying multiple layers) and intra-map (spatial, proximity, neighborhood and region) operators typically desired in a GIS, but the full expressive power of LISP is availabale as well. In the Map Algebra [Tomlin] and many other raster-modelling languages (and image-processing in general), there is an implicit 'For all Rows; For All Columns; Do...' nested iteration construct, which, for example, creates a new map as a sum of two input maps by simply adding each pair of cells in like locations in the input maps to produce the value in the output map. This is usually hard coded, in Pascal, Fortran, C, etc., and provides a useful and reliable base for forming cartographic models. In pseudo-code, each of the n-ary (inter-map, 'overlay or combination') operations may be characterized by the following algorithm:

```
FOR I := 1 TO NROWS
  FOR J := 1 TO NCOLS
    NEWMAP [I, J] := F (INPUTMAP1 [I, J], INPUTMAP2[I, J], ...)
```

where the function F maybe any of a dozen or so standard operations such as sum, product, min, max, etc.

Each of the unary (intra-map, 'spatial, neighborhood or proximity') operations may be characterized by the variant algorithm:

```
FOR I := 1 TO NROWS
  FOR J := 1 TO NCOLS
    NEWMAP [I,J] := F (NEIGHBORHOOD (INPUTMAP1 [I, J]))
```

where the function F is as before, and the function NEIGHBORHOOD returns a list of cell-values derived from a variety of standard options (circular or

square neighborhood of specified radius, all cells with similar values or values within a specified range, etc.)

In EMAPS, these two algorithms are built-in, available to the user through a top-level loop which requires only writing a LISP expression which serves as the function F, and which may use any of a number of pre-defined procedures for the NEIGHBORHOOD function. In these functions, each of the indices i,j and the cell-value at [i,j] is explicitly available as a special (global) variable. That is, in EMAPS the reserved words ROW, COL, and VAL may be used in the definition of both F and NEIGHBORHOOD, and will be evaluated by each cell appropriately.

In the object oriented approach, the algorithms become some variant of:

```
FOR I := 1 TO NROWS
  FOR J := 1 TO NCOLS
    (ASK NEWMAP
         (ASK (CELL  I J)
              (SET-VALUE  (F  (ASK  MAP1
                                    (ASK  (CELL I J)  (GET-VALUE))) ... )))
```

The last line, written now in pseudo-LISP syntax [1], indicates the hierarchical order of message passing, in which each map asks each cell to set- or get- its value. The function (CELL I J) returns the cell-object at location [i,j]; the function F is still any ordinary LISP expression. In fact, in EMAPS, a variety of macros provide access to a variety of different forms of the function F. For example, to set the contents of MAP3 to the sum of MAP1 and MAP2, the expression is:

```
(ASK MAP3  (UPDATE  (ARRAY-SWEEP  (MAP+  MAP1 MAP2))))
```

MAP+ is just a macro which expands into the longer expression

```
(+ (ASK MAP1 (ASK (CELL ROW COL) (CELL-VALUE)))
   (ASK MAP2 (ASK (CELL ROW COL) (CELL-VALUE))))
```

+ is just the ordinary addition function, except that it has been modified to not produce errors when it encounters a non-numeric value, but simply to return the special value NA, or 'No Value'. (All the mathematical functions have been so modified, so that no errors occur from mixing numeric- and non-numeric cell values; this has the desirable side-effect of propagating 'NoValue' in all mathematical expressions, a useful facility in many models. Having such a special value is a great benefit in many data analysis operations; the ability to simply incorporate one is due to the flexibility of the underlying LISP language.)

UPDATE is a message defined for all maps, which says in effect, "set all your cells to the values indicated."

---

[1] In these examples I have used a simplified 'Object Lisp' syntax, of the form (ask object (expr)) which simply means evaluate 'expr' in the environment of 'object'. If expr is of the form (proc args), then this is approximately equivalent to the Common Lisp Object System (CLOS) syntax: (proc object args), assuming proc has been duly defined as a method specialized for object [Keene].

ARRAY-SWEEP is the term in EMAPS for the iterative construct in the algorithms above. (While it is the most commonly used general case, the alternative SPREADING-ACTIVATION function, discussed below, is also available.)

The various standard map operations are all available or easily written as LISP functions: for example, the operation COVER has the following form:

```
(DEFINE-MAP-FUNCTION COVER (MAP1 MAP2)
      (ARRAY-SWEEP
            (LET ((VAL1 (ASK MAP1 (ASK (CELL ROW COL) (GET-VALUE))))
                  (VAL2 (ASK MAP2 (ASK (CELL ROW COL) (GET-VALUE)))))
             (IF (= VAL1 0) VAL2
                            VAL1))
```

Without either assuming or explaining all of the niceties of LISP syntax, suffice it to say that the above procedure defines a map operation which takes two maps as input, and generates a new map such that, for each cell: if the value in the first input map is non-zero, that value is put in the new map, and if the first map value is zero, then the value of the second map is put into the output map (literally 'covering' the second map with the first, where zero is 'transparent')

In the above examples, we have seen ROW and COL used as special variables, successivly bound to all combinations of I,J in the iterative construct, and used only as arguments to the (CELL) function to specify a particular cell. But they can be used in any legal LISP expression, including simply standing alone:

```
(ASK MAP1 (UPDATE (ARRAY-SWEEP ROW)))
```

will generate a map which has all 1's in the first row, 2's in the second row, and so on (numerically, this might represent a tilted plane, tilted down toward the user, ranging in elevation from to nrows.)

Similarly, the special variable VAL can be used, as shorthand for

```
(ASK (CELL I J) (GET-VALUE));
```

the expression

```
(ASK MAP1 (UPDATE (ARRAY-SWEEP (SQRT VAL))))
```

will take the square root of MAP1, i.e. replace each cell value with its square root. Finally, we may even use a constant instead of any variables:

```
(ASK MAP1 (UPDATE (ARRAY-SWEEP 7.5 )))
```

will produce a 'constant' map, all of whose cells have the value 7.5.

All of the invariant part of the expression (ASK MAP1 (UPDATE (ARRAY-SWEEP ... can be wrapped up inside 'syntactic sugar' [Abelson & Sussman], and replaced by the expression (SWEEP ...) (or even placed on an interactive menu system, so all the user has to type is the essential part of the expression.) Thus, the command (SWEEP (+ ROW COL)) will produce another 'tilted plane' map, this one tilted from northwest to southeast, ranging in elevation from 2 to (+ nrows ncols).

The neighborhood functions are also easily defined: the square neighborhood of eight cells immediately adjacent to (CELL I J) is represented by the list

```
( (CELL (1- I) (1- J)) (CELL (1- I) J)  (CELL (1- I) (1+ J))
  (CELL I (1- J))                       (CELL I (1+ J))
  (CELL (1+ I) (1- J)) (CELL (1+ I) J) (CELL (1+ I) (1+ J)))
```

(This is true for an interior cell 1< I < NROWS, 1< J< NCOLS; some proper strategy must be defined for handling of edge conditions.) If the function (SQUARE-NBRS I J R) returns the list of values in the square neigborhood of diameter R (R ODD, R>2), then the expression
(SWEEP (/ (SUM (SQUARE-NBRS ROW COL 3)) 8)) will result in 'smoothing' the map, replacing each cell with the average of its eight nearest neighbors (summing their values, and dividing the sum by eight). A variety of other smoothing and 'image processing' operations such as convolutions can be simply defined in the same way.

Circular or square neighborhoods are the commonest and most easily implemented, but the cellular-automaton model suggest that we might want to be able to specify variants. Indeed, for some purposes it may be desirable for cells to be 'wired' in completely arbitrary connections. In the object-oriented approach, this is easily accomplished: each may mat optioonally specialize the generic 'neighbors' function(s), to return a specialized list of neighbors in each case. The highlevel (SWEEP ...) command remains the same, but the result will depend on the details of the neighborhood interconnections. For example, in some smoothing operations, certain cell values should be considered fixed and not be smoothed; these should simply have their neighbor function redefined to consider only themselves (or, more preciseley, to return their value multiplied by 8, since the smoothing function above divides by 8). More speculatively, we might imagine different kinds of cells, some of whom ignore their neighbors altogether, some of whom are influnced by a relatively small radius, others by a larger radius. A community of such cells could provide a model of certain kinds of peer-pressure in neighborhoods, or variations in species sensitivity to pollutants.

We may also wish to model cells as 'emitters', rather than 'receivers', where the diameter of the neighborhood of influence is not determined by the receiving cell, but rather by the emitting cell. In these cases, and in other scenarios, it is useful to have an alternative to the ARRAY-SWEEP approach which must consider all NROWS X NCOLS cells. If, for example, some cell represents an object which has just emitted a cloud of radioactive smoke, an iterative process which starts there and spreads according to some algorithm away from the source might be more efficient than an array sweep (so long as it is known that distant effects are nil).

For these kinds of models, EMAPS provides the (SPREADING-ACTIVATION ...) macro, which takes as its argument a single cell-position or list of positions, and runs the 'inverse neighborhood' function for each source cell ("In whose neighborhood is this cell included?") iteratively (or recursively). Thus, if there is a global vector (such as a wind speed and

direction) that determines neighborhoods, a plume effect can be modelled; if there is a concentric decay function, that too can be modelled. In this approach, EMAPS continues to cycle until all neighbor lists come up empty, or off the edge of the map, effectively providing an animation of the model.

Hill climbing (or descending) algorithms are also easily implemented in this same style. In fact, in the object-oriented approach, we simply introduce a new kind of object - a CLIMBER. This object interacts with maps and cells, querying surrounding cells for elevation values, choosing a path, and leaving a trail. Whether the climber is responsible for any graphics to be displayed in the window, or cells are responsible for marking the passage of the climber, is a responsibility-allocation (design) question that has no single correct answer. One benefit of the modularity of the object-oriented design approach is that neither maps nor cells must change with the introduction of new types of objects; on the other hand, once those objects are recognized as useful parts of a system, some system redesign may be called for.

CONCLUSIONS

Environmental design and planning (the set of disciplines including architecture, landscape architecture, urban design and planning among others) demands the construction and testing of explicit models of complex and dynamic systems. Traditionally, most often, those models have been constructed in the form of drawings or physical scale models, and tested by visual inspection. These models are eprforce static and over-simplified. Computer-assisted techniques for design and planning models at the larger-than-site scale, epitomized by 'GIS' systems, suffer from much the same limitations. These systems for managing and manipulating spatial information, whether in vector- or cell-based format, are designed primarily to manage multiple overlays of descriptive data and produce analyses of a statistical nature; the integration of these analyses into a synthetic design or planning process is left outside of the system.

Some models are now routinely computed in the course of environmental design -- stuctural and thermal, financial, economic and other calculations performed in stand-alone applications or general-purpose spreadsheets are common. The integration of these models with drawing (CAD) or mapping (GIS) systems, especially across different scales of resolution and abstraction, is ad-hoc at best, impossible or incomplete much of the time, and fails dramatically at worst. Some solutions have been proposed and attempted: inter-application data communication is an area of intense research and development, based largely on the definition of standards.

The approach advocated here is not meant to deny the benefits of standards and inter-process communication, but describes an alternative in the form of an integrated environment for design and planning that extends the domain of grid-cell based data. The declaration of object boundaries, names and class-instance relationships is a major part of design knowledge, especially in schematic and preliminary design stages.

Object-oriented programming systems that provide static object descriptions and hierarchies, and that restrict or forbid dynamic restructuring of links, will not serve in a design environment, in which the ability to add and delete parent- and child-links at any time is an essential part of design development and model exploration.

The structure and behavior of EMAPS is also particularly conducive, I believe, to the development of an approach to "Designing with Maps". Designing with Maps requires a different approach than either traditional GIS or CAD systems: more flexible, speculative and prescriptive than GIS; more abstract and geographic than CAD. A traditional description of designing has it composed of analysis, synthesis and evaluation; EMAPS supports the synthesis stage better than traditional GIS modelling packages which were designed primarily for analysis and evaluation. As part of an ongoing inquiry into models of computer aided design, including maps and other modes of representation, EMAPS is being interfaced with other sofware in a heterogeneous network. LISP is especially conducive to this sort of integration, as it enables the control of other applications and operating systems from within a single, flexible programming environment.

EMAPS is particularly valuable for modelling and exploring dynamic interactions in landscape ecology, and it should be useful for teaching basic concepts of programming geographic information sytems. In conjunction with other software and systems, such as for three-dimensional renderings and animations, EMAPS provides the kind of flexible, extensible, systematic modelling 'testbed' required for the next generation of computer-aided environmental design.

REFERENCES

Goldberg, Adele & David Robson, 1989. *SmallTalk-80 The Language,* Reading, MA: Addison Wesley Publishing

Hogeweg, P. 1988. "Cellular Automata as a Paradigm for Ecological Modeling", *Applied Mathematics and Computation* 27:81-100 (1988)

Itami, R.M. 1988. "Cellular Automatons as a Framework for Dynamic Simulations in Geographic Information Systems", in *Proceedings, GIS/LIS,* American Society for Photogrammetry and Remote Sensing, Falls Church VA 1988, pp. 590-597.

Keene, Sonya. 1989 *Object-Oriented Programming in COMMON LISP: A Programmers Guide to CLOS.* Reading, MA: Addison Wesley Publishing

Steele, G.L. Jr. 1990. *Common Lisp The Language,Second Edition,* Bedford, MA: Digital Press, 1990

Toffoli, T. and N Margolus, 1987 *Cellular automata machines : a new environment for modeling.* Cambridge, Mass. : MIT Press, 1987.

Tomlin, D. 1990. *Geographic Information Systems and Cartographic Modelling,* Prentice-Hall, 1990.

ACKNOWLEDGMENTS:

The development of EMAPS has been supported by an equipment grant from the Milton Fund of Harvard University and by the excellent software development environment offered by Macintosh Common Lisp.

FEATURE INFORMATION SUPPORT AND THE SDTS CONCEPTUAL DATA MODEL: CLARIFICATION AND EXTENSION

Leone Barnett & John V. Carlis
Department of Computer Science, University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

The Spatial Data Transfer Standard (SDTS) includes a definition of spatial objects and spatial features presented as part of what the standard calls a "conceptual model." Using a precise data modeling notation, we developed a Logical Data Model (LDS) of the SDTS spatial object. Unfortunately, the SDTS description of features, and their relationship to spatial objects, is too ambiguous to model precisely. However, once an interpretation of key concepts is assumed, that information can be modeled. We offer our interpretation, and use that as the basis for an extended model of real world features and their relation to spatial objects. The analysis and the resulting model are useful for designing a system that more effectively manages the "meaning" of its spatial objects.

INTRODUCTION

The Spatial Data Transfer Standard (National Institute of Standards and Technology 1992) presents a conceptual model of spatial objects, real world phenomena, and their relationship. We attempted to produce a data model of this with Logical Data Structure (LDS) notation, which is a precise specification that can be mapped directly to system implementations. The spatial object description was clear enough to model directly. However, the sections describing features and real world phenomena require some interpretation before a model can be specified. Therefore we present two LDS models: one reflects the SDTS spatial object description, and the other offers an extended view of the relationship between spatial objects and the real world features they represent. Our extended view makes certain types of information explicit, which is important for developing systems that will be capable of more sophisticated spatial data management. We use LDS notation for our models, so a brief introduction to it is provided next.

A Logical Data Structure (Carlis 1993) is a graphical diagram that names types of data and relationships to be remembered. An LDS is formed from basic structures called entities, attribute descriptors, relationships, relationship descriptors, and identifiers. Data modeling with LDS diagrams is a technique that supports precise data definition and clear communication of what data types are of interest. An LDS entity is displayed as a box, with its name in the top section. An LDS entity names a type of data. An LDS entity is described by a relationship descriptor when there is a relationship between the entity and another LDS entity. An attribute of an LDS entity appears as text inside the box for the entity. A relationship, depicted as a line connecting two boxes, may be one-to-one, one-to-many, or many-to-many. This indicates, for example in the one-to-many case, that one instance of entity-1 has relates to zero or more instances of entity-2.

Because the SDTS uses the term "entity" in a sense that is not defined the same as the "LDS entity," we must distinguish our usage. For this paper, when we mean to refer to "one of the boxes in an LDS model" we say "LDS entity." When we leave off "LDS" as a qualifier, we are using the term in the conceptual sense of the SDTS, or in our interpretation of same.

THE SDTS

Overview of SDTS

The Spatial Data Transfer Standard (SDTS) is a collective effort to standardize the transfer of spatial data. It has been designed for the transfer of data that are structured and used by scientists who use spatial data. According to USGS Fact sheet, Aug. 1992,

it "provides an exchange mechanism for the transfer of spatial data between dissimilar computer systems." The SDTS was approved as a Federal Information Processing Standard (FIPS) 173, in July 1992.

The SDTS was developed to facilitate the sharing and transfer of spatial data across systems and users. As such, the concepts, ideas, and structures in the standard strongly reflect current systems and commonly used models for spatial data. To be generally relevant, it attempts to relate to GIS in general. However it is not a trivial problem to develop a standard that has general applicability and acceptance when use of terms and agreement on the best ways to conceptualize spatial issues is lacking. Spatial data models have long been an important topic in the GIS community, but terminology associated with the various models is not always consistent.

For example, Peuquet (Peuquet 1990) discusses two types of spatial data models most commonly used in computer systems. She calls them vector and tessellation models. Vector models use points, lines, and polygons to represent real world features or concepts (such as rivers, contour lines, political boundaries, etc.). Tessellation models involve methods of breaking up space into pieces, which may have regular, irregular, or nested divisions. Egenhofer and Herring (Egenhofer & Herring 1991) also describe different types of spatial data models, but their two most important groups of models are called regular tessellations and irregular tessellations of space. They describe vector structures as a means of implementing an irregular tessellation, and grid cell raster structures as a means of implementing a regular tessellation. This illustrates that a common term, i.e. "vector," is used to refer to a model in one case, and to a mechanism for implementing a model in another case.

The lack of consistency in how people think about spatial data models is due, in part, to the tendency to mix "what" and "how" together. That is, what is being modeled is mixed with terminology that describes how different structures are used to do the modeling. With spatial models, sometimes space itself is modeled, and sometimes things in space are modeled. Because they both have spatial structure, it is possible, if not always convenient, to use any of the methods for either purpose. For example, vector models are commonly, if not always, used when the primary task is the representation of real world entities (especially geographic ones). The real world entities have a position in space, but the space itself is assumed rather than modeled. This necessitates explicit modeling of spatial characteristics of the real world entity, often referred to as its topology. In addition, the location of the real world entity modeled might be viewed as an attribute. Regular tessellation models, such as the raster grid, are used when a primary task is the modeling space. Space is modeled by breaking it up into contiguous pieces. Qualities associated with the space, or things contained in the space, are attached secondarily as attributes of the space.

The SDTS does not resolve or even address these differences. Rather, it describes a spatial object that is based on the most commonly used and understood spatial models - the geometric or vector model, and the raster type of regular tessellation model. The spatial object description they offer for the vector model is quite detailed, and it was relatively straight-forward to develop an LDS that illustrates their description. The level of detail in their spatial object description reflects the historical and still prevalent, strong focus on geometric representations of real world phenomena, rather than on the phenomena themselves. This is due to the obvious utility of spatial descriptions for map-like displays and for storage of spatial data.

The detail of the SDTS spatial object description stands in contrast to the level of detail used to describe another concept, which they refer to as the "entity object." This concept focuses on the use of the spatial object for representation of real world features, or on the connection between these two, and is a concept that is separate from the spatial object descriptions per se. Their interest in the real world features, as well as their lack of a detailed description, reflects the current and growing interest, but lack of consensus, in how to develop "feature-based" GIS. Such systems promise to provide more of what may be called "semantic information" support. This refers to the system's capability of a) knowing more about data (what they mean, how they are represented,

etc.); b) being able to use the data (specialized operators are available); and c) being capable of making decisions as part of processing (interpretive rules are available). We contend that understanding and developing support for the "entity object" is central to enabling a system to capture important semantic information. Because of this, the SDTS spatial object description and entity object description are both examined closely in the sections that follow.

The Spatial Object: Geometry and Topology

The SDTS provides a detailed description of a spatial object, and the development of an LDS graphical data model for this part of the SDTS description was relatively unproblematic. The concept of the spatial object, as described in the SDTS, is precisely modeled with the LDS shown in Figure 1. The next few paragraphs describe the model depicted in this figure. As with the SDTS spatial object description, the most important aspects are the geometric and topological structures.

The description of a spatial object containing the most detail is one that is based on vector spatial data models. The most basic type of spatial object is the point. Figure 1 shows that the LDS entity "Point" is a kind of 0-dimensional object that is uniquely identified by the attribute "NP," which is a point id. It has "x," "y", and optionally "z" attributes that are coordinates used to specify location. The "point-type" attribute is used to indicate the role of the point, which may be as a point feature location, a text display location, or an area feature attribute handle.

Other types of spatial objects are made up of points. A "Line-segment" is a pair of unordered points. A "String" is a sequence of points. An "Arc" is a locus of points that form a curve defined by a math expression. These are all 1-dimensional objects. A "Polygon" is a 2-dimensional object with an "Interior area" and a boundary. A polygon can be a "GT-Poly" or a "G-Poly" which differ with respect to what kind of boundary they have. A "GT-Poly" has a "GT-Ring" for a boundary, which is a closed "Chain" sequence. A "G-Poly" has a "G-Ring" for a boundary, which is a closed "String" or "Arc" sequence. The substantive difference in the two polygon types is whether or not they carry topological information. The "GT-Poly" does and the "G-Poly" does not.

Two kinds of topological data are modeled. One, an adjacency relationship, is modeled via the entity called "Chain-Poly pair elt" (or element). This entity captures the idea that a "Chain" has a relationship with one "GT-Poly" on its right, and another "GT-Poly" on its left. A "GT-Poly" is either on the left or right of each of the chains comprising it. This left or right position is called "face." Since face (or left-ness and right-ness) depend on orientation, a second kind of topological data, direction, must be captured, which is modeled in the LDS entity called "Chain-node ordered elt." A "Chain," comprising a sequence of line segments or arcs, also has two nodes. These nodes are ordered as the first and last, or beginning and end. Direction of a chain moves from the first node to the last node, with the chain's sequence falling in between. Since one node may be associated with more than one chain, the many-to-many relationship between chains and nodes is modeled with "Chain-node ordered elt."

These structures all relate closely to what Peuquet calls the vector spatial data model, which uses geometry to describe the shape of something, and possibly topology to record some spatial relationships. The shapes occur in space, but the space containing them is not specified directly. Rather, it is implicit, and at best derivable. At worst, there is no explicit mapping to, for example, a location on earth. In contrast, a regular tessellation spatial data model makes the description of the containing space explicit. The SDTS offers a limited description of spatial objects which are based on the regular tessellation model. The standard specifies the "Pixel" and the "Grid cell" as 2-dimensional spatial object primitives that can be aggregated into contiguous sets called either a "Digital image" or "Grid" respectively. Both of these are types of "Raster" data. With these spatial objects, the containing space is explicitly defined, but shapes, objects, or other descriptions of the things in space must be derived.

Despite the lack of equal attention given to all types of spatial models (others are

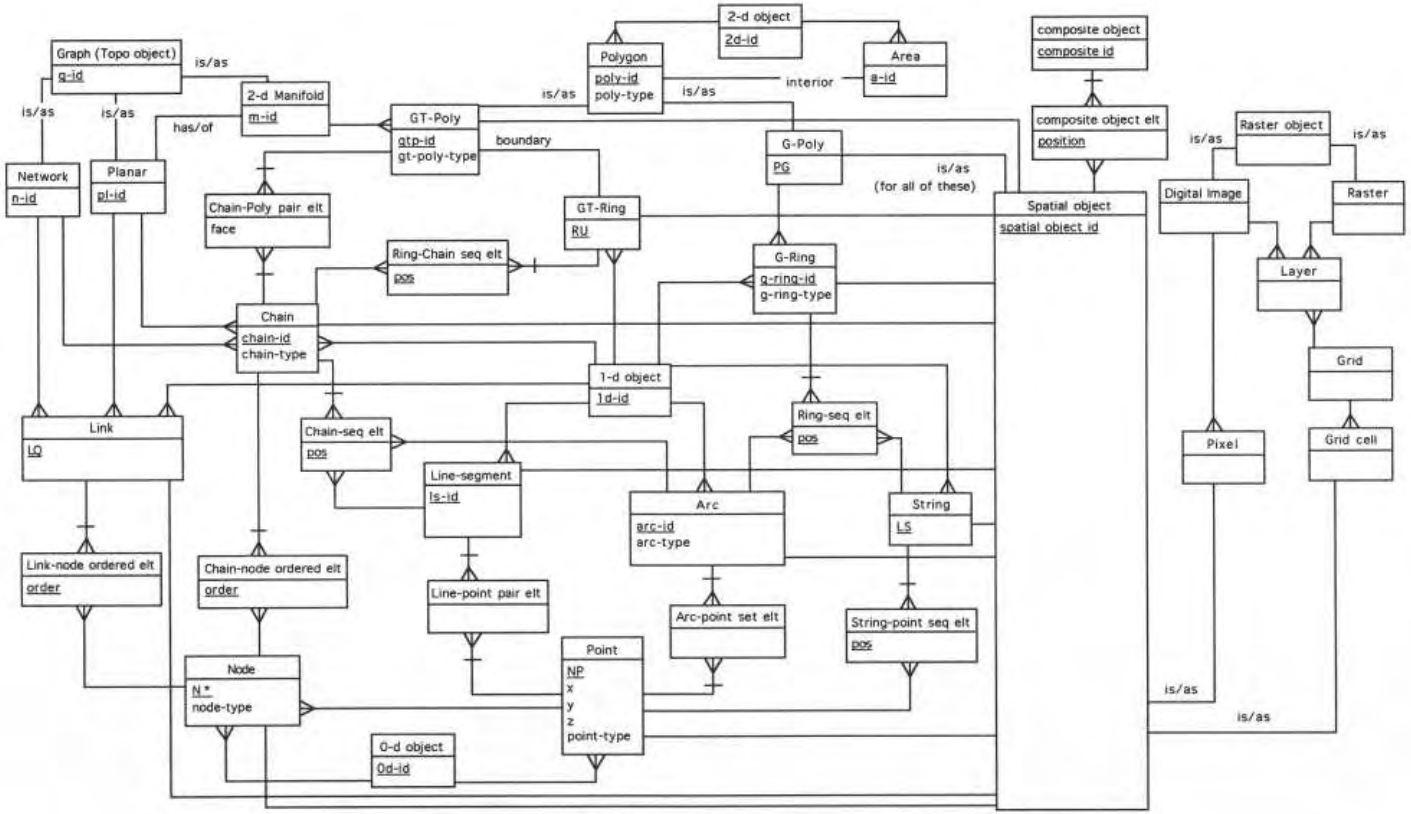

Graph (Topo object)
g-id
is/as
is/as
is/as
2-d Manifold
m-id
has/of
Network
n-id
Planar
pl-id
GT-Poly
gtp-id
gt-poly-type
boundary
Chain-Poly pair elt
face
GT-Ring
RU
Ring-Chain seq elt
pos
Chain
chain-id
chain-type
1-d object
1d-id
Link
L0
Chain-seq elt
pos
Line-segment
ls-id
Arc
arc-id
arc-type
String
LS
Link-node ordered elt
order
Chain-node ordered elt
order
Line-point pair elt
Arc-point set elt
String-point seq elt
pos
Node
N *
node-type
Point
NP
x
y
z
point-type
0-d object
0d-id
2-d object
2d-id
Polygon
poly-id
poly-type
Area
a-id
is/as
interior
is/as
G-Poly
PG
is/as
(for all of these)
G-Ring
g-ring-id
g-ring-type
Ring-seq elt
pos
composite object
composite id
composite object elt
position
Spatial object
spatial object id
Raster object
is/as
is/as
Digital Image
Raster
Layer
Grid
Pixel
Grid cell
is/as
is/as

Figure 1

described by Egenhofer and Herring), the SDTS made an important statement. The standard treats vectors and rasters (and potentially other structures) as conceptually similar things in the sense that they are all spatial objects of one type or another. This emphasizes that there is a common ground in that they all deal with the structure of either space or things in it. Also it helps build recognition of the difference between structural spatial concepts and other concepts (such as attributes) that are not inherently spatial at all. That is to say that the structure of space, or things within a space, has different modeling requirements than the properties of the things that occupy space, or the abstract descriptions of features of the space.

Given the spatial models that it covers, the SDTS spatial object description is considerably detailed, and relatively straight-forward; and it stands in contrast to the description they offer of real world phenomena that spatial objects presumably represent. This is the topic of the following section.

Real World Phenomena: Confusion and Ambiguity

The SDTS is primarily concerned with spatial object data transfer, but recognizes that the context of use of spatial data involves real world features, or phenomena. The standard addresses this aspect of spatial data in two ways. First, it presents what it calls a "conceptual model" that serves as a "framework for defining spatial features and a context for the definition of a set of spatial objects." Second, it provides a set of common definitions for terms which are types of real world phenomena. In this section, we examine the "conceptual model" of the SDTS which offers a description of real world phenomena and spatial objects.

The SDTS does not provide a description of a model of the real world phenomena at the same level of detail and completeness as is provided for the spatial object description. This is not surprising since the development of spatial representations has been a dominant focus of research, whereas attempts to codify aspects of the represented phenomena, spatial and non-spatial, is more recent. This newer interest is due, in part, to requirements for spatial analysis support that involve richer models of the phenomena represented. The lack of detail and ambiguous description of concepts concerning the real world phenomena makes it difficult to clearly interpret the standard with respect to this topic. We examine two problem areas and, for each, justify an interpretation of the standard that is reasonable, although not the only one possible.

Explicitly Distinguish Real World from Model The first problem is that the SDTS does not explicitly state a distinction between real world phenomena and a model of what a system must remember about same. It is important to make this distinction because there is a difference in discussing phenomena themselves vs. the things about phenomena that must be remembered or recorded as data. In defining concepts of real world phenomena, where a "river" would be a class of phenomena, and the "Mississippi River" would be an example member of that class, the SDTS says, "... classes of phenomena are called entity types and the individual phenomena are called entity instances." Although this implies a discussion of modeling things to be recorded or remembered about phenomena, it could be interpreted in two different ways. One interpretation is that an "entity type" is an abstract class of phenomena and an "entity instance" is actually the phenomenon itself, which means the Mississippi River itself is an instance of the entity type River. A different interpretation matches the following re-phrasing of the SDTS statement: "In a data model, those named classes of phenomena that are to be remembered (and presumably recorded in a computer system) are called entity types and the named individual phenomena of a given type are called entity instances." This asserts that an "entity type" names a class of phenomena. Likewise, an "entity instance" names a particular member of the class. "River" is a named type, and "Mississippi River" is a named instance. The instance refers to the real world river, but clearly is not the river itself.

The latter interpretation is actually more useful for a discussion of data representation and standardization, and it is reasonable to assume that the standard can be interpreted this way. This is supported by the standard's use of concepts and terminology that

derive from data modeling and knowledge representation literature. For example, the SDTS defines classification, generalization, aggregation, and association in terms of entities and entity instances. These are the same names of specific concepts that are often discussed in data modeling (in database and other computer science literature), and they are discussed in a very similar manner (Brodie & Mylopoulos 1986, Peckham & Maryanski 1988, Hull & King 1987, Sowa 1984).

However, there is a possibility that the former interpretation is the one that was intended. Deciding which interpretation is intended depends on how the issue of an "entity instance's" dependence on spatial objects is resolved. This topic is addressed next.

Entity Instance Dependency on Spatial Objects It is difficult to determine what level of dependency an "entity instance" has on a spatial object. The SDTS says that "entity instances have a digital representation (that) consists of one or more spatial objects." This raises the question of whether or not entity instances can be modeled and digitally represented independent of spatial objects. Despite this potential "entity instance" dependence, the independence of spatial objects is implied fairly strongly. The SDTS says that spatial objects do not have to represent an entity instance, but if they do they are called "entity objects."

The standard goes on to say that an "entity object (is) a digital representation of an entity instance," which, in the context of the discussion, may lead to the conclusion that the only kind of digital representation for an entity instance is in the form of a spatial object. It is easy to find that this constraint, if enforced, is overly limiting. For example, text may be used to represent entity instances, or their attributes, and GIS currently handle at least some such data as text. However, if one does assume that entity instances depend on spatial objects for existence, one might also assume that "entity instance" means the real world phenomenon itself, and an "entity object" is the representation of it using some spatial object as the means of description.

Whether or not it is the intention of the SDTS to assume the independent existence of entity instances (by which we mean the independent modeling of real world features), this independence is necessary for designing systems that can support semantic information relating to real world phenomena. This idea is elaborated in the next section.

The Entity Object Concept

In this section, we review the "entity object" concept introduced in the SDTS. To understand and model this concept precisely, it is first necessary to determine what the SDTS means by "entity instance." Once an interpretation is provided, a meaning of the "entity object" can be proposed.

We assume an interpretation wherein the SDTS "entity type" and "entity instance" are used to model (and name) real world phenomena, and that "entity instances" are not, strictly speaking, the phenomena themselves. We assume that models of real world phenomena may involve both spatial and non-spatial data, but they may exist as models independent of spatial models that can be related to them. Given these assumptions, the "entity object" described by the SDTS models the relationship between an instance of an entity (which models a type of real world phenomenon) and a spatial object used to represent it. Justification for the assumption of entity instance independence, and the subsequent interpretation of the entity object as a relationship follows.

Assuming the Independence of Entity Instances Viewing the entity object as a relationship necessitates viewing entity instances as having separate existence from spatial object representations, since, in general, a relationship between two things implies the separate existence of the things to be related. Below we argue for the existence of an entity instance independent of a spatial object.

An entity instance must be either a real world phenomenon or a modeled instance. As a real world phenomenon it clearly has a separate existence. As a particular modeled

instance, it may or may not exist separate from a spatial object representation. However, it is clear that it is possible to model things about real world phenomena that are not spatially oriented, and it may indeed be useful to do so. For example, rivers are wet, they may be muddy, they have current and direction, they support eco-systems, etc. In addition, it is useful to recognize when two different spatial objects actually refer to the same real world phenomenon. System support for recognition of such an occurrence is simplified if real world phenomena are modeled independently of their spatial representations. Thus it is useful, in the general case, to assume the option of independent entity instances, even if in many cases (i.e. actual system implementations) there is no separate existence or model.

The advantage of modeling entity instances in a manner that supports their independence is that this allows the system to be able to deal with concepts that apply to real world phenomena that are in addition to, and independent of, a geometric view of them. The need for this kind of view is reflected by the increasing interest in GIS that handle additional data about real world phenomena besides the spatial data. Albeit that spatial objects are used to represent the real world phenomena, and often this is the only representation of them the system has access to, in order to be more informed about them, the system must have access to more complete or different models of them.

In sum, a spatial object representation should be one, but not the only, representation of real world phenomena available to the system. Richer representations of the meaning of spatial objects in the context of real world phenomena require greater flexibility in modeling these things. In addition, the system ought to be able to support multiple spatial object representations for the same entity instance.

Viewing the Entity Object as a Relationship The standard implies that an entity object involves some sort of relationship between a spatial object and an entity instance without specifying the details. There are indications to support that an entity object actually is a relationship, even if the standard does not state this expressly. (Remember that part of the trouble is that without establishing the clear independence of entity instances and their role in modeling, it is hard to discuss the entity object as a relationship.)

The SDTS does say that an entity object is a spatial object that represents an entity instance. Since an entity instance ought to have an existence that is independent of a spatial object representation, and since spatial objects apparently do have independent existence, this implies that an entity object exists when these two are paired together. That is, an entity object exists when a relationship exists between spatial object and entity instance.

The standard has taken a step toward clearly separating spatial objects from the phenomena they represent (i.e. the entity instances) in another way. They explicitly acknowledge the confusion resulting from overloading the term "feature", which can mean either the real world phenomena, or the geometric (spatial object) representations of same. Because of the different ways the term is used, the standard specifies that the term "feature" means a spatial object only when it is used to represent a real world phenomenon.

This distinction can be made more clearly and effectively, by defining an entity object as a relationship between two independently defined concepts. That is, the entity object is the relationship between a particular entity instance (which models some particular real world phenomenon) and a particular spatial object. Then, a feature may be defined as "the combination of the real world phenomena that is modeled as an entity instance, and the spatial object used for its geometric representation." With this definition, a "feature" is effectively an "entity object" which in turn is a relationship.

EXTENDING THE MODEL

In this section we address some limitations of commonly used models for GIS, by describing and extending a model based on the SDTS. The purpose of the extensions is

to make more data available to the system, enabling it to make more decisions about data management that are based on a greater understanding of the meaning of the data stored, i.e. the semantics. Semantic information includes domain specific data concerning real world phenomena, as well as specialized information on the use of geometric data, its properties, conversions, and interpretation. Our extensions center around the concepts of the real world phenomena that spatial objects may represent, the spatial object itself, and the relationship between these things.

The Meta Model Concept

The SDTS presents a description of a context, which they call "a conceptual model," within which their description of spatial objects can be understood. We can view the SDTS "conceptual model" as a statement about a meta-model of GIS, wherein a model of the types of things you create models for in a GIS would be a meta-model.

Specifying that, in general, concepts about real world phenomena are of interest, without specifying exactly which classes are of concern, is a meta-modeling concept. In a meta-model, generality of the structures used to model phenomena is important because the various views of real world phenomena, and what users want to remember about them, can vary greatly from application to application. In contrast, a specific class of phenomena, such as roads, is an application level concept, which would involve a data modeling effort concerning use of roads data in a specific GIS application.

We use an LDS to describe a meta-model of GIS, based upon but extending the SDTS. The meta-model is shown in Figure 2. The model reflects general components of GIS but includes data that would allow a system to handle greater complexity and semantic interpretation. The motivation for decisions reflected in the model are the subject of the next three sections. We organize the discussion around three principle concepts: the modeling of real world phenomena themselves, the spatial object used to represent modeled instances, and the relationship between these two.

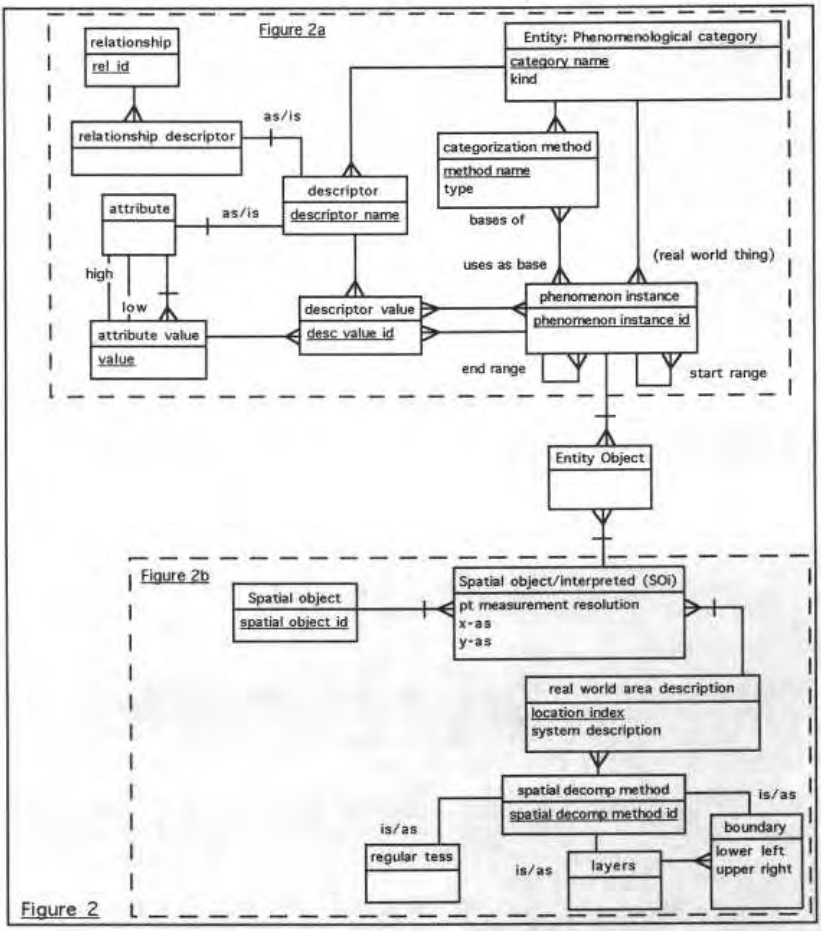
Real World Phenomena and Attributes

In this discussion of extensions to the GIS model, we focus on developing the model of real world phenomena and their attributes. The important points made are that clear, explicit separation of the real world phenomena model is necessary, and general capability for modeling real world complexity is important.

Spatial Object Model and Real World Phenomena Model Separation Clear separation of the model of spatial objects and the model of the types of real world phenomena that a GIS might support is illustrated in Figure 2. LDS entities that model real world phenomena are grouped so that connections to spatial objects occur at the entity object. Justification for this type of clean separation is that real world phenomena are complex and much of the complexity is irrespective of any spatial object representation. This separation is necessary for both handling multiple spatial object representations of the same phenomena, and for designing systems that can manage complexity and multiple views of the data.

Complexity of real world phenomena The SDTS asserts the existence and complexity of the real world phenomena and suggests that we can model them with various constructs. The LDS fragment in Figure 2a (a part of Figure 2 enclosed with a dotted line box) expresses our interpretation of what is required to adequately model real world phenomena, and to do so in a way that does not assume spatial object dependency. As a meta-model, this fragment is similar to meta-models previously developed, and found useful in general database design (Carlis & March 1984, and Held & Carlis 1989).The LDS entity called "Entity: Phenomenological category" is what the SDTS calls an "Entity." We added the phrase "Phenomenological category" to emphasize that instances of this are used for the modeling of real world phenomena. "Entity: Phenomenological category" has descriptors which are either of the type "attribute" or "relationship." Relationship descriptors allow many kinds of relationships (including class/subclass, etc.) to be created between phenomenological categories that are modeled

in an application LDS. An instance of an "Entity: Phenomenological category" has many "phenomenon instances" which represent real world features or phenomena, such as the "Mississippi River." These "phenomenon instances" have associated with them many "attribute value" of "attribute" 's.



Kinds of real world phenomena Part of the complexity concerning real world phenomena is that things concerning or describing the real world are of different kinds, including at least physical, political, and abstract. The meta-model we present offers a unified concept for dealing with the fact that in general things occupy space, things have attributes, and attributes are used to describe different areas. An important aspect of this model is that it allows the system to manipulate an abstract phenomenological instance in a manner similar to a physical phenomenological instance. An abstract phenomenological instance of something like "an area of land use that is agricultural," can be treated in a way similar to a phenomenological instance of a river. Why are these similar? Because both occupy space in some sense, but both are separate from a specific location. Some instances may in fact shift over time, in terms of the location they occupy. In addition, both a river and a homogeneous, contiguous area (such as an area that has an agricultural land use), have in common the tendency to be described by geometry.

In order to provide a general model of the real world phenomena modeled in GIS, we

need to allow an "phenomenological instance" to include abstract descriptions of an area, such as land use. Although a land use value, such as "agricultural" is actually an attribute associated with an area, we can model this as a phenomenological instance of an abstract kind of phenomenological category. The utility of this view is that an area with special properties attributed to it can be manipulated and given a geometric form in a manner conceptually similar to a physical concept. And they both are viewed by the system as occupying space. We can still maintain clear separation of location and things, even abstract things, that are in that location.

A separate issue is how classes get created, and have instances assigned to them which ultimately relate to spatial objects.

Examples of Types and Instances Since meta-models are abstract and difficult to understand, examples of how the meta-model relates to some specific data models of real world phenomena are presented in Figure 3. First, note that there is a difference in the meta LDS and the application level LDS. In a specific application, such as a database or GIS built for some purpose, an LDS contains LDS entities that name the types of data you want to remember and store instances of. In the meta-model (in our case, of GIS) the LDS contains LDS entities that name arch-types in a sense, or "types of types" of data. Thus, instances of "Entity: Phenomenological category" in the meta-model are names of LDS entities in a GIS application LDS.

Figure 3 shows some simple example data. The "mapping" between a meta level LDS and application level LDS's is illustrated with instance data. We show that an instance of "Entity: Phenomenological category" at the meta level becomes the name of a type (i.e. an LDS entity) in a application LDS. The instance of "Phenomenon instance" at the meta level is an instance of an application level LDS entity. The examples include classes that are physical, political, and abstract.

The Spatial Object

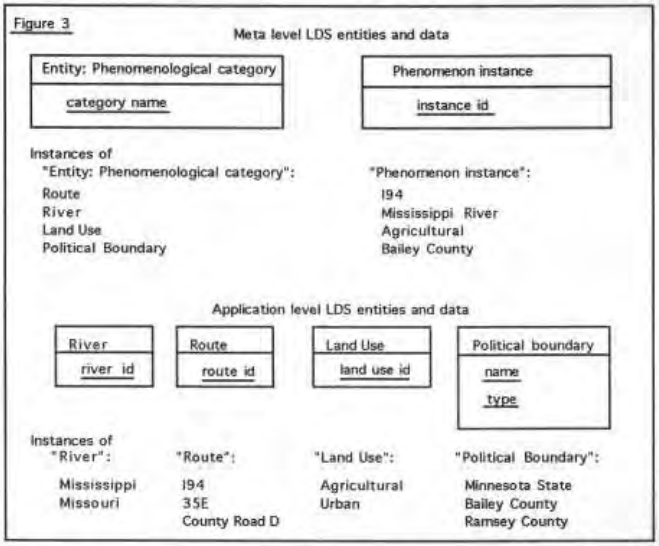
This section describes extensions that support an explicit understanding of the spatial object in the context of real world location. It introduces the notion of an interpreted spatial object, where the word "interpretation" is used to signify how a spatial object maps to a real world location, and how the basic building blocks of spatial objects (points or cells) actually map to an area. Note that in the meta-model in Figure 2, we show the spatial object as an LDS entity. However, for simplicity, we do not include the additional detail (i.e. all LDS entities that relate to spatial object) shown in Figure 1.

Separate concept of location from geometry of something in the location One purpose of a spatial object in a GIS is to use it as an indication or specification of a real world area or location. As a spatial object is a geometric description based on organization of points, or a raster like collection of cells, it can be mapped to many real world area descriptions. A "real world area description" is shown as an LDS entity in the LDS fragment identified as Figure 2b (a part of Figure 2 enclosed with a dotted line box). A real world area description may have many spatial objects that map to it. This creates a many-to-many relationship between a spatial object and a real world area description. When a spatial object is mapped to a real world area description, so that the x and y values of the points in the spatial object map to a particular area in the real world, we call it an "interpreted spatial object."

Introduce Measurement Resolution Because a geometric point is a pure abstraction, with no area, we introduce the concept of measurement resolution associated with a point to provide more explicit interpretation of how much real world area the point is used to represent. When a spatial object based either on geometric notation or raster cell models, is mapped to a location, there is an explicit or implicit assumption concerning the measurement resolution of a point or cell involved. This is because the spatial object is not used as a purely geometric object, but rather to describe some object set at some location.

We first discuss how measurement resolution relates to a raster cell. A raster cell is used

to stand for an area rather than something in an area, but it is typically associated with data collected and tied to an area. That real world area has some dimension in the sense that there is a minimum or average distance between points at which the data were collected. This can be calculated as a length and stored as "pt measurement resolution." For raster cells, this value gives a measure of the real world area associated with a single cell.



Less intuitive is the notion of measurement resolution when applied to vector or point based geometric data. However even in this case, a line, for example, is used to represent a real world feature such as a road. There is an assumption, although implicit, of width of the road. Thus the point ought to be interpreted in a manner that conveys the notion of width. "Pt measurement resolution" can be used with geometric points to indicate a real world area that the geometry supposedly relates to.

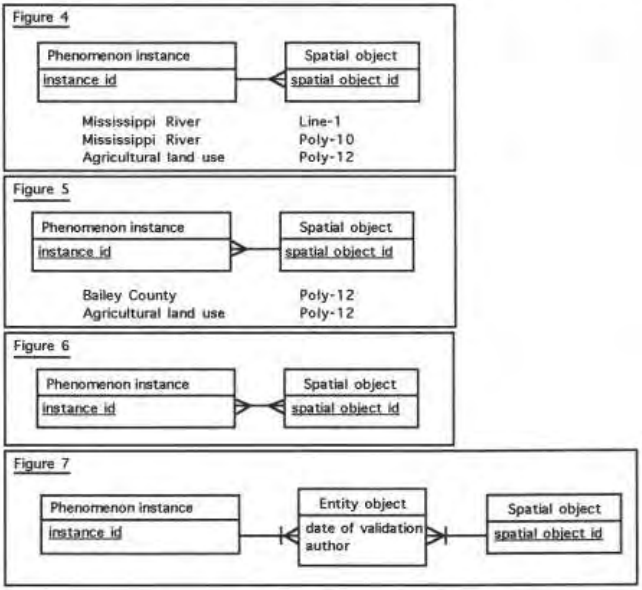
The result is an interpreted spatial object which has a geometric or simple cell description, as well as a means for specific mapping to real world location. It is the interpreted spatial object that is really the basis for describing and representing real world phenomena, and thus the meta LDS in Figure 2 models this. However, the interpretive aspects are often managed by users rather than a GIS. With a meta-model expressing this explicitly, we can explore more options for developing system support of this type of information. This clearer specification of what is really meant by a spatial object can enable a system to know about, and subsequently be able to use, information in more sophisticated data processing and information management.

Modeling The Entity Object Relationship

In this section we focus on the SDTS's entity object concept. The SDTS's description of the "entity object" expresses some sort of relationship between a "spatial object" and an "entity instance." We explore the nature of this relationship, with sufficient attention to detail to allow modeling it with the LDS notation. Different ways of viewing the relationship, which may be investigated as part of a modeling process, result in different ways of representing the entity object concept in an LDS. The important messages here are that the entity object, being essentially a relationship between two complex things, is actually quite complex itself. Other data, which must be modeled, relate directly to this union

Entity Object Complexity The entity object concept introduced in the SDTS

should be examined in terms of the different ways in which relationships between spatial objects and phenomenon instances might exist. Figure 4 shows an example where a single phenomenon instance ("Mississippi River") has many possible spatial objects ("line-1" and "Poly-10"). The LDS models this possibility. Another situation, modeled in Figure 5, shows that a single spatial object ("Poly-12") may be associated with many phenomenon instances ("Bailey County" and "Agricultural land use"). The general case, in Figure 6, is that phenomenon instances may have many spatial objects, and a spatial object may represent many different phenomenological instances.



In addition to being a many-to-many relationship, the relationship itself may have descriptors, which, when using LDS notation, means the relationship is promoted to an LDS entity itself. Once this happens, specification of types of data that relate specifically to the relationship is possible.

In a meta-model, we are interested in modeling the most general case, thus Figure 7 models the entity object as an LDS entity allowing it to have its own descriptors. We presented this logic up to here using the "spatial object " instead of the "interpreted spatial object " for simplicity. As is the case with many real systems, this simpler model of a spatial object implies that locational interpretation, if desired, is provided by the user. The extension of Figure 7 that shows how the entity object relates to the interpreted spatial object is shown in Figure 2.

Entity Object as a Concept Occupying a Locati The entity object concept concerns the modeling of an object that occupies space at some real world location. It is not the real world phenomenon itself, nor the attributes accorded to a location, nor the location itself. But location of the object (including the abstract objects such as a land use coverage) is determinable along with the phenomenon.

UTILITY OF THE EXTENDED MODEL

As real world phenomenological information becomes a more important aspect of GIS, the standard may be expanded to include transfer of such data. Better models of these data (which ultimately would be the basis for a system's understanding of what real world features are, and how they are related to spatial objects) are needed for extending the standard in this direction. A step toward this is conceptual analysis that results in a

clearer data model of the types of information involved.

The extended model highlights the role of the entity object. Seeing the real world phenomena and spatial object distinction clearly, facilitates analysis at the more conceptual level, removing the need to focus on data structure. The model is based on a separation of *what* the data are from *how* they are stored or represented in the system. This type of modeling activity helps us to identify new types of data to use in the processes that enable system decision making. Fuller, richer distinctions, modeled and made available to the system, are part of the process of semantic information analysis. If the system is to handle more of the data management responsibility, more complete modeling of the actual meaning of data must be made explicit, and not stored only in people's minds. The system cannot use what it has not been told about.

Clear separation of what types of data actually are used and how they ought to be interpreted will be increasingly important as more data are collected and must be managed by systems.

CONCLUSION

The SDTS contains ambiguous statements concerning their conceptual model of real world "features" and their relationship to spatial objects. Disambiguation is necessary for development of a clear model of these data, and is important if systems are to be designed that will effectively manage such data. Assuming one reasonable interpretation of the standard, an extended model of important types of GIS data is proposed, and presented in LDS notation. The model makes data explicit that typically are implicit, and potentially available to humans, but not to systems. The model also illustrates the relationship between real world entities and spatial objects in a manner that differentiates them clearly. This facilitates better modeling of real world phenomena and supports development of future GIS systems that will have a greater ability to manage the "meaning" of the geometric (spatial object) data. This will allow development of specialized processing in spatial analysis that is closely tied to, or dependent upon other information besides spatial object information. It will also allow development of systems than can handle more of the data integration burden that GIS commonly cannot handle, due to their insufficient "knowledge" of the meaning of spatial objects that are stored.

REFERENCES

Brodie, M.L., and Mylopoulos, J., 1986, *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*, Springer-Verlag, New York.

Carlis, J.V., 1993, *Logical Data Structures*, Book manuscript in press.

Carlis, J.V. and March, S.T., 1984, "A Descriptive Model of Physical Database Design Problems and Solutions." In *Proceedings of the IEEE COMPDEC Conference.*

Egenhofer, M.J., and Herring, J., 1991, "High Level Spatial Data Structures for GIS," from D.J. Maguire, M.F. Goodchild, and D.W. Rhind (eds.), *Geographic Information Systems*, Vol. 1, pp. 457-475, Longman Group UK, Harlow.

Held, J. and Carlis, J.V., 1989, ADM-An Applicative Data Model, *Information Sciences*, January, 1989.

Hull, R., and King, R., 1987, "Semantic Database Modeling: Survey, Applications, and Research Issues," *ACM Computing Surveys*, Vol. 19, No. 3.

Peckham, J., and Maryanski, F., 1988, "Semantic Data Models," *ACM Computing Surveys*, Vol. 20, No. 3.

Peuquet, D.J., 1990, "A Conceptual Framework and Comparison of Spatial Data Models," in *Introductory Readings in Geographic Information Systems*, Donna Peuquet and Duane Marble (eds.), Taylor & Francis, New York.

Sowa, J.F., 1984, *Conceptual Structures*, Addison-Wesley, Reading, Massachusetts.

GEOGRAPHIC REGIONS:
A NEW COMPOSITE GIS FEATURE TYPE

Jan van Roessel and David Pullar
ESRI
380 New York Street
Redlands CA 92373
jvanroessel@esri.com
dpullar@esri.com

ABSTRACT

ARC/INFO 7.0 will have a new capability to handle overlapping and disjoint areas through a new feature class: the "region." A region consists of one or more non-overlapping areal components. Regions may overlap. Two relationships are maintained between the composite region feature and the base polygonal and chain data: (1) a region-polygon cross reference, and (2) a region-boundary chain list. Regions can be interactively created and edited, or may be constructed in bulk from arc loops. Regions are maintained by other GIS functions such as overlay. Coverages resulting from overlay inherit the region subclasses from the parent coverages. Spatial inquiries can be made against multiple region subclasses within the same coverage through "regionquery" and "regionselect" functions.

INTRODUCTION

Many GIS users are concerned with the management of overlapping and disjoint areas of interest in a single coverage of a vector-based GIS. Often because of the frequency of overlap, and the irregular way in which it occurs, it has not been practical to manage overlap by using different coverages.

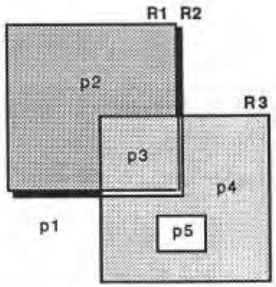
Application areas that can benefit from managing irregular overlapping data in a single coverage are varied and many. Oil and gas applications must keep track of overlapping lease data. They are also concerned with overlapping geological data at various depth levels. Forestry applications must manage stand treatment data, fire histories, and other historical data. Nature conservation deals with natural communities, plant and species habitats, which are not spatially mutually exclusive. Cadastral and parcel mapping must keep track of overlapping historical parcels and legal land records. AM/FM applications may want to manage different floor plans in the same coverage. Applications that have nested data at different levels, such as Bureau of the Census data, are also a prime candidate for an implementation using regions.

The development of the region feature class results primarily from the need to manage overlapping data, but the term region is more associated with areas that may be spatially discontiguous. The new feature class also provides the capability to manage noncontiguous areas with identical attributes as a single region.

BACKGROUND

In the past many ARC/INFO users have implemented some type of scheme to deal with overlapping data by implementing systems using the Arc Macro Language (Gross 1991). The most sophisticated system of this type has been a cadastral-parcel mapping system developed by ESRI-UK. Invariably, all these schemes have used the device of a cross-reference file. This file stores the relation between the overlapping units and the base polygons. Figure 1 show the use of a cross reference file

With the introduction of routes and sections in ARC/INFO 6.0 it became clear that a similar composite type could be constructed based on polygons. Such a feature type would provide "core" support for the overlapping polygon problem, freeing users from having to implement custom made schemes.



Region	Polygon
1	2
	3
2	2
	3
3	3
	4

Figure 1. Cross-reference file example

The basic design question was whether to implement an approach where geometry is shared, or whether to create overlapping polygons with duplicated geometry. Both options are actually available for lines in ARC/INFO. The non-planarity of lines is achieved in two ways: (1) logically through the route and section system, by which multiple coincident routes may coexist with shared geometry, and (2) with duplicated geometry through the existence of coincident arcs and crossing arcs without nodes at the intersection.

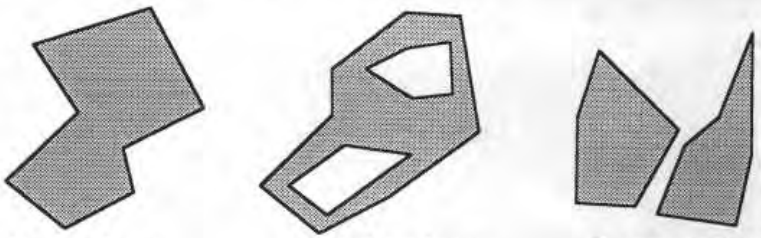
The same options were available for polygons. We selected the shared geometry approach, with multiple subclasses. This is equivalent to the route and section system. One important reason for choosing the shared geometry approach is cadastral applications, where the basic geometry must not change when parcels are subdivided.

LITERATURE REVIEW

It is apparent that the term region has many definitions in geography. Therefore, it is inappropriate for us to give a formal or encompassing definition for this term. Rather, we concentrate on where region is defined as a spatial unit in a GIS.

The spatial units used conventionally to represent discrete geographic entities are points, lines, and polygons. Spatial units are classified based upon their dimensionality and geometric properties. Laurini and Thompson (1992) define spatial units for polygons and regions. The definitions, see figure 2, are as follows;

- a simple polygon is a connected 2-dimensional space bounded by lines,
- complex polygon is a connected 2-dimensional space bounded by lines containing one or more holes,
- region is made up of two or more separate simple polygons that may be disjoint.



Simple polygon
(no holes or exclaves)
[cell]

Complex polygon
(has exclaves)
[area]

Non-connected polygon
(has exclaves)
[region]

Figure 2: Polygons and regions from Laurini and Thompson

Laurini and Thompson also describe spatial units from the perspective of how they are combined. In particular they define a compound type as an object created by combining spatial units of the same type. A combination of polygon objects forms a new compound type that has different semantic properties than the single unit. For instance, a compound spatial type for land parcel may be composed of a polygon for the parcel boundaries and a polygon for a house located on the parcel.

Their definition for region is reasonably consistent with the region feature type in ARC/INFO. The region is used to represent areal entities in 2-dimensional space as a non-connected polygon, but an ARC/INFO region is allowed to have exclaves. Regions are also consistent with the definition for a compound object. A region is a composite made up from the underlying polygons, but it has its own semantic properties. A region has descriptive properties that are independent of the properties for its component polygons.

A more formal definition for regions is given in Scholl and Voisard (1989). A region type has both spatial domain properties and set-theoretic properties. That is, a region type can be considered as the spatial domain for an object, and it also signifies that a region type is a composite made up from a set of elementary subsets of space (which are also regions). Here an elementary region is defined as a subset of R^2. From figure 3 we see it can be any of the following:

- a polygon (e.g. r3, r4)
- a subset of R^2 bounded by lines (r1, r2)
- the union over R^2 of several connected or non-connected parts (e.g., r3 $\cup$ r4)

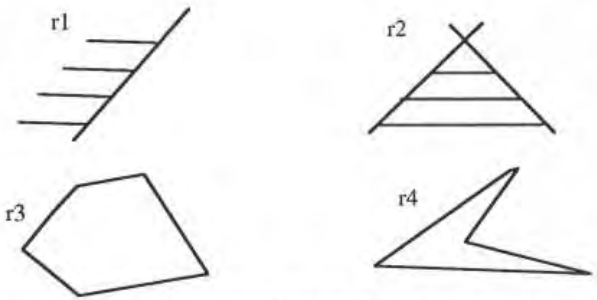


Figure 3. Regions as defined by Scholl and Voisard

This definition of regions by Scholl and Voisard is too general. The regions in ARC/INFO use a stricter interpretation. The subsets of R^2 that are part of a region must be 2-dimensional. Formally we say that the subsets of points have a 2-dimensional neighborhood. This interpretation recognizes the fundamental difference between objects that have a 1-dimensional neighborhood, i.e., linear objects, and those that have a 2-dimensional neighborhood, i.e., areal objects.

ARC/INFO Data Model

The ARC/INFO data model is composed of a number of geographic data sets: coverages, grids, attribute tables, tins, lattices, and images. The coverage data set encompasses several feature classes. Depending upon the user's method of abstracting real world entities, geographic objects are represented as points, lines, or areas. These object types can be modelled in an elementary way as nodes, arcs, and polygons (Morehouse 1992). Elementary types are related in a topological sense to form a partition of a planar surface (similar to the USGS Digital Line Graph). One coverage is then analogous to a surface layer. Combinations of elementary geometrical types are made to build more elaborate representations of geographic objects. These are called composite feature classes. They include sections, routes and regions. The sections and routes feature classes are used to

build route systems on top of arcs and nodes to define dynamic models over a network. Regions are a new feature class that provides a similar capability as routes, but build arbitrary areas on top of elementary polygons.

Figure 4 is a schematic of the relationships between feature classes in a coverage data set. The figure shows that composite feature classes are built upon elementary feature classes. This allows users to build integrated coverages as combinations of the elementary types. Structurally a region is composed of many polygons and the set of arcs that form the exterior boundary of the region. The arcs are related to nodes by the encoded connectivity relations (to/from), they are related the polygons by the encoded coincidence relationships (left/right), and may also be related to a route system encoded as composite relationship.

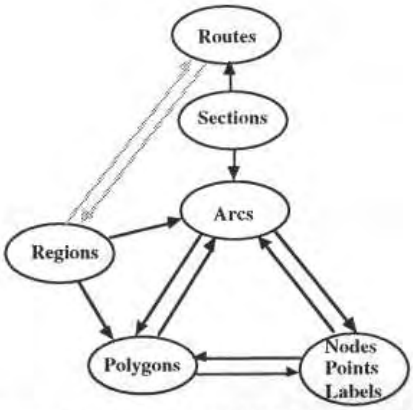


Figure 4. Regions and ARC/INFO Data Model

The relationship between polygons and regions can be understood by the ownership ordering for areal spatial units. A hierarchical tree graph is used to illustrate these relationships (Laurini and Thompson 1992). Figure 5 shows how regions and polygons can be modeled as a two tier graph.

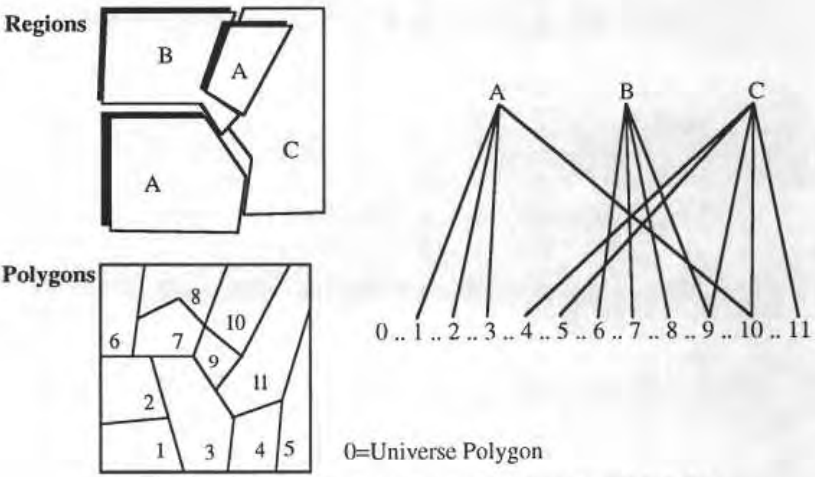
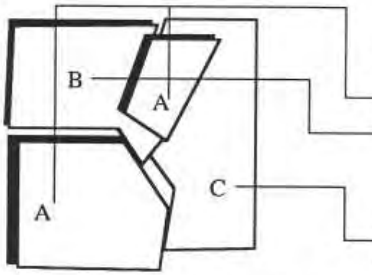


Figure 5. Regions and polygons modeled as a two tier graph

The ordering is a many-to-many and there is no nesting. The lowest level has the elementary spatial units, namely polygons. These units are aggregated to form a composite spatial unit, namely a region, as arbitrary combinations of polygons. Note that polygons partition the plane into a universe polygon and then 1-N elementary polygons. Regions are composed of subsets of the N polygons, but the universe polygon cannot belong to a region. Each region borders on the universe for both internal and exterior boundaries, and a hole within a region belongs to the universe. This allows boolean logic operations on regions that cannot be expressed on a polygonal partition in ARC/INFO where each connected component is always covered by non-universe polygons.

Another important aspect of the ARC/INFO data model is that each feature class is associated with an attribute table. For elementary feature classes there are separate feature attribute tables for nodes, arcs and polygons in a coverage data set. Composite feature classes allow several attribute tables to be associated with sections, routes, or regions. For this reason the individual attribute tables for composite classes are referred to as subclass tables. The way a user organizes a coverage is to have one composite feature subclass for each homogeneous set of attributes. In other words, a set of regions with common attributes is assigned to the same region subclass. For example, within an integrated coverage one region feature subclass may be used to represent types of forest with cover attribute information, and another region feature subclass may be used to represent flood hazard areas with flood level and date attribute information. For each region instance in a composite feature there is one record in the subclass table. Figure 6 shows a schematic of how regions are related to attribute tables.



Forrest Subclass

Region	Area	Cover
A	100.1	hardwood
B	110.5	pine

Flood-Harzard Subclass

Region	Area	Level	Date
C	121.7	20.5'	09-06-89

Figure 6. Region subclasses and attribute tables

The storage structure for regions uses three files, a PAL file to store the region-arc relations, a RXP file to store the region-polygon relations, and a PAT file to store region attribute information. The first four fields of the PAT file are maintained by the system, they include the record number, user identifier, area, and perimeter. Since there may be many region subclasses within a coverage a naming convention is adopted. The PAT file is uniquely identified as **<cover>.PAT<subclass>**, the PAL and RXP files follow this convention.

FUNCTIONAL MODEL

The functionality of the ARC/INFO system resides in a number of functions operating on coverages. Different actions occur based on the feature types present in the coverage. With spatial features such as regions and polygons new spatial features may result. Newly generated polygons must always be assigned to a new coverage, because polygons are mutually exclusive. The unique overlap quality of regions does not pose this constraint, so that regions may stack up in the same coverage.

ARC/INFO's GRID system similarly has the concept of "stacked" grid data, but one of the unique differences between rasters and regions is that the depth of the stack is uniform for a grid, but it may be variable for regions.

The functional mode for regions takes advantage of the fact that regions may accumulate in a coverage.

Let $Cov(A)$ denote a coverage with a set of region subclasses A. F_b is a binary operator that produces one output coverage from two input coverages. The region functionality for this type of function is:

$$Cov3(C) = Cov1(A)\ F_b\ Cov2(B)$$

where $C = A \cup B$. If A and B contain an identical subclass s, each with region sets P_S and Q_S, then C will have subclass s with a region set $R_S = P_S \cup Q_S$. With identical subclass, we mean that the subclass bears the same name and has the same attributes, and therefore has the same schema. The function will fail for subclasses with identical names and different attributes. Examples of F_b are the ARC/INFO functions UNION, INTERSECT and IDENTITY.

A second type of binary function is G_b, with the model:

$$Cov3(A) = \mathrm{Cov1}(A)\ G_b\ \mathrm{Cov2}(B)$$

where the region subclasses from the second cover do not enter into the output cover. Examples of this type of function are CLIP and ERASE.

Another class of functions is the unary function $F_u(A)$ operating with single coverage input and output. Here the subclass set A has a single subclass s. This function produces regions in subclass s, and has the overall effect:

$$Cov2(C) = F_u(A)\ Cov1(B)$$

where $C = A \cup B$. Examples of this type of function are REGIONQUERY REGIONBUFFER. For instance, in the case of REGIONBUFFER, B may be an empty set and buffer regions are produced from points, arcs or polygons into output subclass s.

For functions of type F_u, A and B may have a subclass with an identical name, but different attributes. The function will not fail, as with F_b. Denoting a subclass schema of subclass s by $\{name, I\}$ where I is a set of attributes, then if A contains a subclass s with $\{sname, I\}$ and B contains a subclass q with $\{sname, J\}$ then C will have a subclass r with $\{sname, K\}$, where $K = I \cup J$.

For a function of type F_b, attributes rows of regions in subclasses with the same schema are simply appended. For a function of type F_u, attribute rows are appended, inserting blanks or zeros, as appropriate, into the values for the attributes I.

CONSTRUCTING REGIONS IN BULK

Regions can be constructed in a number of ways. They can be created interactively, or they can result from operations on other regions, or may be created in bulk, starting with arc data.

A special function named REGIONCLASS builds "preliminary regions" out of arc data. The preliminary regions produced by the REGIONCLASS function groups a set of arcs based on a user specified attribute. Each group of arcs must form one or more rings for one region. The rings are constructed and recorded in a region PAL file.

The preliminary regions are then converted into fully built regions in the CLEAN process. The arcs are first intersected and then enter a planesweep polygon construction phase in which polygons are built from intersected line segments. Each line segment has a set of associated regions that is propagated in the planesweep process. This produces the region cross reference file for each subclass. The cross reference files are then used in turn to update the region PAL files for each subclass. This completes the process of building topologically structured regions.

EDITING REGIONS

Regions with the ARC/INFO ARCEDIT system can be edited as a collection of arcs or as a collection of polygons. ADD and DELETE commands add to the current region, or

delete from the current region the primitive feature's type selected. If arcs are edited the selection commands require the user to select the region(s) by pointing to the arc(s) that belong to that region. Similarly, if editing polygons, the selection commands require the user to select the region(s) by pointing to the polygon(s) that belong to that region.

When the coverage is saved, each region subclass is updated to reflect the changes. In either case, the PAL file is updated to reflect the new arc and node numbering scheme. If polygon topology is maintained, then the RXP file is update to reflect the new polygon numbering.

REGIONS IN OVERLAYS

Coverages that are the result of overlays inherit the region subclasses of the input coverages. For the UNION, INTERSECTION and IDENTITY functions, the output coverage receives the subclasses from both input coverages. For the ERASE and CLIP functions, region subclasses from the erase or clip coverages are not inherited. For an ERASE, the area of the ERASE cover is blank, and for a CLIP, only the outline of the CLIP cover is used for clipping.

The extent of regions appearing in the output coverages is limited by the extent of the overlay result. This is shown in Figure 7. As a result of the overlay operation, subclasses may become empty, but they do not vanish.

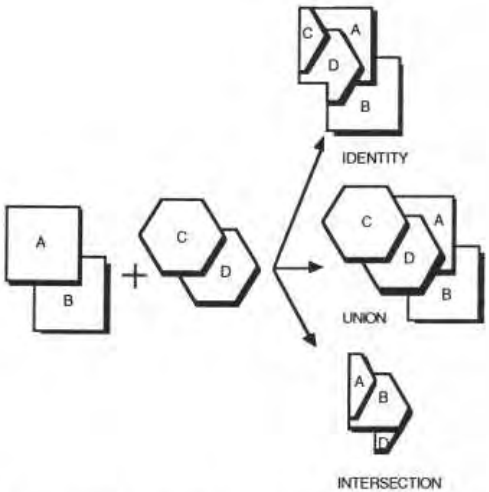


Figure 7. The classical overlay operators and their effect on regions.

The region subclass inheritance mechanism for overlays revolves around a fast updating of the subclass RXP files, using the polygon parentage files produced by the overlays. If an input polygon P has an associated region set R, and another input polygon Q has an associated region set S, and P intersects Q, then the region set for the intersected polygon is the union of the region sets of the parent polygons. This is illustrated in Figure 8. The updated RXP files are then converted into updated region PAL files.

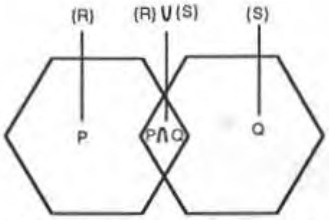


Figure 8. Region set associated with intersected polygons

One way in which regions may be used is to do "integrated coverage" analysis. Instead of keeping the various polygonal thematic layers in separate coverages, they can be integrated into a single coverage as region subclasses. Analysis and queries are then performed with much greater speed in the integrated coverage, provided of course that the base data do not change. Some precision may also be lost due to fuzzy tolerance processing for the integration. An easy way to integrate two polygon coverages is to copy the polygons of each input coverage into regions using the POLYREGION command. This is then followed by a UNION overlay. The resulting integrated cover will show the input coverages as region subclasses.

REGIONS AND "DISSOLVE"

The counterpart of intersecting polygons in overlays is merging of polygons in functions such as DISSOLVE. In that case, if we have a polygon P with region set R and a polygon Q with region set S, and P is combined with Q into a single polygon, the region set for P $\cup$ Q becomes R $\cap$ S.

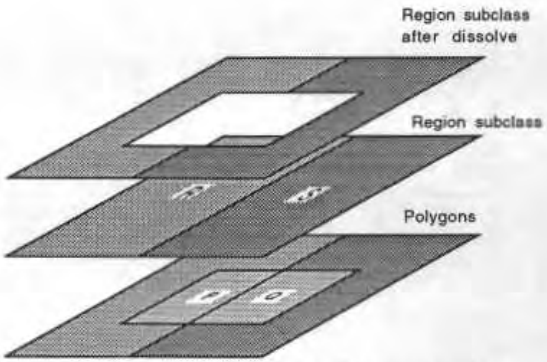
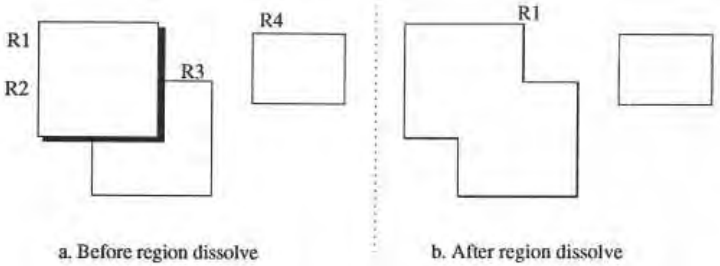


Figure 9. Effect of merging polygons on regions

An example is shown in Figure 9, where polygons P and Q have single associated regions R and S. Merging polygons P and Q yields an empty region intersection set, so that the region subclass after the dissolve develops a "hole."

The above example shows how traditional polygon dissolving has region implications. It is also possible to merge and dissolve regions. Dissolving polygons merges adjacent polygons that have the same value for a specified item. Because regions can be discontiguous, adjacency plays no role when dissolving regions. The following example shows how regions are dissolved. Assume a region subclass with regions shown in Figure 10a.



a. Before region dissolve b. After region dissolve

Figure 10. Region dissolve example

Regions R1 and R2 are two identical overlapping squares. Assume further that the subclass feature attribute table has an attribute that has the same value for each region. Using REGIONDISSOLVE with this attribute creates one output region as shown in Figure 10b. Note how the dissolve not only occurs in the horizontal dimension, but also takes place vertically by removing overlap.

ANALYSIS WITH REGIONS

While the initial impetus for implementing regions was to keep track of, and manage overlapping areal units, it became clear that regions may be used for unique types of analysis. At the moment we have implemented two functions, REGIONQUERY and REGIONSELECT, but there are other opportunities that may be exploited in the future.

REGIONQUERY is an ARC function that can mimic the ARC/INFO "classical" overlay functions in an integrated coverage. But this is just one of its capabilities. While the "classical overlay functions" are two at a time functions, REGIONQUERY can perform boolean selection against any number of region subclasses. In addition to selection, it also produces output regions that are homogeneous with respect to a number of user specified output items.

REGIONQUERY usage can be represented as:

<out_subclass> [WITH <in_subclass.attribute...in_subclass.attribute>]

WHERE <logical_ expression>

where *<out_subclass>* is the output subclass, pre-existing or to be created, *<in_subclass.attribute>* is an attribute of *<in_subclass>* that will be added to the schema of *<out_subclass>*. The set of these attributes forms the output attribute list. The list is optional.

The *<logical_expression>* is a completely general boolean expression where the operands are either constants or attributes of the form *<in_subclass.attribute>*. A special pseudo item *$subclass*, meaning "where the subclass exists" (no-zero record number) can also be used. The logical expression used defines the "territory" qualifying for the query, while the output attribute list defines the "granularity" of the output. Territory in the region's case also extends to the vertical dimension, and the same is true for granularity, because identical overlapping regions with identical output attribute values are collapsed to a single region (see the previous "dissolve" discussion).

The REGIONQUERY function also has an option to produce contiguous output regions.

The model under which REGIONQUERY operates is the following. Each polygon in the input coverages has a number of regions for a given subclass of which it is a part. These are the regions "above" the polygon. The polygon and the regions above it form a subclass "stack."

If only a single subclass is involved in the query, the selection expressions are evaluated for each member of the stack, and if true, the polygon is selected to be a part of a candidate output region corresponding to that member of the stack for which the expression is true.

If multiple subclasses are present, and a single polygon has multiple subclass stacks with more than one region, REGIONQUERY will combine the stacks in the form of a cartesian product, and evaluate the attributes according to the logical expressions for each member of the product.

Output items are assigned to each selected cartesian product combination and a dissolve is performed to make the attribute value combinations unique for each output region.

Example 1

This is a traditional site selection example. The objective is to select a proposed laboratory site where the preferred land use is brush land, the soil type should be suitable for development, the site should be within 300 meters of existing sewer lines, beyond 20 meters of existing streams and be at least 2000 square meters in area.

Assuming an integrated coverage with a land use, soils, stream buffer and sewer buffer subclasses we can pose the following query using the contiguous region option:

potential_sites WHERE landuse.lu-code eq 300 and soils.suit >= 2 and not $streambuf and $sewerbuf

This is followed by a second query using the *potential_sites* subclass:

qualifying_sites WHERE potential_sites.area > 2000.0

Example 2

A typical problem concerning regions or polygons is to find out which units of one class are completely or partially contained in another class. Containment is related to overlap, where 100% overlap means fully contained in (or a duplicate of) and 0 % overlap means the opposite. In general one can ask the question display all units of class X that are p% contained in class Y, where 0 <= p <= 100.

Assuming that we have an integrated coverage with "floodplain" and "vineyards" region subclasses we can solve the problem of selecting all vineyards that are at least 80% contained within the floodplain with two queries:

overlap WITH vineyard.vineyard# WHERE $floodplain and $vineyard

This creates regions that are the intersection of the floodplain and vineyard subclasses in a new subclass called overlap. They will have as an attribute the record number of the vineyard region (vineyard.vineyard#).

Then we do the following query:

mostly_within WHERE (vineyard.vinyard# = overlap.vineyard#) and
(overlap.area / vineyard.area >= 0.80)

Here we use the unique region overlap property to compare the areas of two regions that share a common overlap area. The output regions in the "mostly_within" subclass are those vineyard portions that overlap the floodplain and are at least 80% of the original vineyard size.

Example 3

Given that we have a set of overlapping lease data stored in a region subclass called "leases" that has an attribute "leascholder" we can generate a report of overlap pairs by first making a copy of the lease region subclass, and name it "leasecopy." Then we run the following query:

conflict WITH lease.leaseholder leasecopy.leaseholder
WHERE lease.leaseholder <> leasecopy.leaseholder

When REGIONQUERY is confronted with one or more subclass region stacks over a polygon, it makes the cartesian product and guarantees that the output will have a region for each unique overlap combination where the leaseholders are different. The attribute

table of the output subclass "conflict" will contain a unique combination of two different leaseholders that have overlapping leases in each row.

REGION DISPLAY AND SELECTION

The ARC/INFO ARCPLOT REGIONSELECT function is a companion function to REGIONQUERY. Unlike REGIONQUERY, it does not create new output regions. The user specifies output subclasses to which the selection refers, rather than output attributes. Like REGIONQUERY, the user also specifies a logical selection over multiple region subclasses. Regions in the specified set of subclasses that have attribute values in their overlap area for which the logical expression is true are selected. As in REGIONQUERY, polygons may be used as another region subclass.

One use of REGIONSELECT is to identify all regions of which a selected polygon is a part. REGIONSELECT operates in the ARCPLOT selection environment, and therefore interacts with past selections.

REGIONSELECT may be also be used in place of REGIONQUERY, if the output regions are identical to already existing regions. For instance, in the second example above, REGIONSELECT may be substituted for the second query.

Regions can be displayed similarly to polygons. A REGIONLINES command can be used to display regions with offset lines, so that overlapping regions that are identical in shape and size can be differentiated.

There may also be other as yet unexplored display techniques that would more effectively communicate impressions of overlapping areal data to a user. A model might be used in which region shades have gradients across a region, so that a region's identity is uniquely shown in a transparent display model.

CONCLUSION

We believe that the ARC/INFO 7.0 release will have well rounded initial capability to handle disjoint and overlapping data. However, there may be requirements and uses that will only become clear when the new feature class is applied in practice.

It is also interesting to speculate how regions may fit in with future GIS development. A potential application may be in the area of fuzzy GIS data (Heuvelink and Burrough, 1993). The traditional polygonal model represents a prismatic probability density function with a 100% certainty that the attribute occurs within the prism contour. With a fuzzy data model overlapping regions may be used to store various levels of a probability density function pertaining to the probability of the presence of an attribute. A corresponding region modeling capability may be needed to manipulate attributes and their associated probabilities.

Another used for the feature class may be in the support of partially ordered sets and lattices (Kainz et al, 1993). Partially ordered sets defined on regions might be used to handle a larger variety of containment queries.

REFERENCES

Gross T. 1991, "Modeling Polygons Located at Several Positions on a Z-Axis as a Single Planar Map." In Proceedings, Eleventh Annual ESRI User Conference, Vol. 2, pp. 5-20.

Heuvelink G.M.B., and Burrough P.A. 1993, "Error propagation in Cartographic Modelling Using Boolean Logic and Continuous Classification." International Journal of Geographic Information Systems, Vol. 7, pp. 231-246. Taylor & Francis, London, Washington D.C.

Kainz W., Egenhofer M.J., and Greasley I. 1993, "Modelling Spatial Relations with Partially Ordered Sets." International Journal of Geographic Information Systems, Vol. 7, pp. 215-229. Taylor & Francis, London, Washington D.C.

Laurini R., and Thompson D. 1992, "Fundamentals of Spatial Information Systems." Academic Press, London.

Morehouse S., "The ARC/INFO Geographic Information System." Computers and Geosciences, Vol. 18, No. 4, pp. 435-441

Scholl M., and Voisard A. 1989, "Thematic Map Modeling." In Proceedings, Symposium on Very Large Spatial Data Bases. L.N.R.I.A., 78153 Le Chesnay, France.

PATHWAYS TO SHARABLE SPATIAL DATABASES

Geoffrey Dutton
Harvard Design & Mapping Co., Inc.
80 Prospect Street
Cambridge MA 02139 USA
01-617-354-0100
qtm@cup.portal.com

Why Johnny Can't Read (spatial data)

Recent advances in data interchange standards for spatial information pay increased attention to differences in data models used to encode such information. This answers a very real need: new spatial information systems continue to emerge into the marketplace, each possessing unique arrays of data models and data structures. As time goes on, users will adopt greater numbers of spatial systems, tailored to particular ends, such as municipal, transportation and resource management, emergency planning, law enforcement, desktop mapping, strategic marketing and real estate applications. Each such system deployed costs money, incurs commitments and populates databases. While digital cartographic data exchange standards such as FIPS 173 (STDS) enable databases to feed one another — perhaps losing information, perhaps not — data remains multiply represented in autonomous archives, with all their overhead, inconsistencies and hassles.

Based on their successes, and given the diverse and decentralized market for spatial information handling, GIS and related technologies seem to have matured to a point where vertical applications are gaining significant market share. In this environment, interoperability is far from the norm, and its prospects are not improving. The best practical solution has been to use *Autocad*™, which in addition to providing a near-universal — if limited — data exchange format, also serves as a data repository for several successful GIS products. But practical as they are, *Autocad* solutions are not appropriate for many applications; GIS vendors have yet to fully accept the imperatives of a client-server world order, as relational database vendors have done, and as personal computer software vendors are starting to do.

Breaking down barriers between spatial databases will require agreements to be reached about many details, including data dictionary standards, descriptions of topology and coordinate spaces, feature encoding schemes, handling of time, and formats for embedded metadata and other conceptual tidbits. This paper discusses several computational paradigms for addressing these and other issues: (1) just-in-time database construction from source files; (2) a universal spatial data repository; (3) a standardized spatial query language; and (4) application programming interfaces to databases. While all of these alternatives are found to have merit, none is seen as sufficient, and each is limited by the complexity and openness of data models it is able or willing to communicate.

Economic and other realities of database proliferation will compel users, researchers and vendors to address how to better share spatial data. Whether suitable, scalable, synoptic standards will surface is uncertain, but few earnest efforts are in evidence. To realize its technical conclusions, this paper advocates building connections between researchers and vendors, bridging applications and data models, and transcending data transfer and metadata debates to build a more universal consensus about access to spatial data. But any of the paths it describes may equally well lead to discovery, amplify applications, stimulate industry and serve users in many ways, if followed deliberately and enthusiastically.

Just-in-time Database Construction

Some GIS data layers and features don't change very much. Political units, administrative boundaries, subdivision plats, zoning areas, roadways, river reaches and topography are examples of lines drawn on maps that change very slowly, so their GIS representations rarely need be updated. Other themes, such as land cover, wetlands, coastlines, utility networks and store locations may transform themselves daily, monthly, seasonally or steadily over time. If a theme requires repeated updating, it may be best to rebuild its GIS representation from source data whenever a project calls for it, and to delete or archive the representation once the project ends. As source data is usually stored in public formats (such as TIGER files or Digital Line Graphs), there are few operational obstacles to sharing it among applications in a client-server environment other than the time required to read in source data and structure its information to conform to a system's data model.

Even themes that are relatively stable tend to be duplicated excessively. Redundant copies of political, census and administrative boundaries abound in GIS databases, propagated whenever a new project may need them. Some projects may need versions of themes bearing specific timestamps, while others may require their official or most recent realizations. In any case, it always is important to have access to authoritative source data, or to well-constructed databases incorporating it. If one's GIS makes sharing project data difficult, new projects need to build themes from appropriate primary data, document what they did to achieve this, and purge databases from secondary storage when they cease to be needed. Activity logs should be kept – preferably in a DBMS – as evidence of the spatial databases built, and can reconstruct them if necessary. Commercial and academic GIS tools are now available that track database lineage and other data quality parameters, as most GISs still do not manage metadata very well. Such tools enable users to clean out entire directories of intermediate results, saving only source data and beneficial output; the rest tends to be less important and can be reformulated if ever it is truly needed.

On-demand database construction is particularly appropriate when source data comes in public formats such as DLG, SDTS and TIGER, and needs to be accessed by various applications. For some of them, just-in-time may not be a preferred strategy; it may be the *only* approach capable of assuring data integrity. Utilities and local governments using AM/FM and GIS in tandem are particularly vulnerable to GIS data obsolescence; in many instances urban infrastructure data is assembled and maintained in CADD environments, then converted into GIS data for thematic mapping, spatial analysis or decision support. Unless a GIS can access primary AM/FM data directly, its themes are in danger of getting out of date and out of synch (schools that have closed, streets without sewers). Always building fresh GIS databases (or at least whenever any sources have changed) is the most reliable way to maintain their logical consistency, although it may not be the least-work way. However, as is discussed next, using a common AM/FM/GIS database is an alternative solution that might avoid much copying and reconstruction, and assures that GIS applications have access to the most current data available.

Universal Spatial Data Repository

As using CADD databases to support GIS activity has proven to be a viable

technology in many milieus, one could argue that a working model for a universal spatial database exists, and it is *Autocad*. In addition to database standards, Autodesk (Sausalito, CA) has provided capable drawing tools and an extensible programming environment that promotes AM/FM/GIS interoperability. Several commercial GISs are rooted in this environment, but also incorporate a DBMS to store topology, feature definitions, attribute data, metadata and their relations (Autodesk is rumored to be teaming with several other vendors to market their own GIS). A large complement of utilities that add GIS value to *Autocad* are also available from third parties. But if *Autocad* is the answer, what is the question?

The main question posed by this paper is *how can any GIS gain access to the highest-quality spatial data in its environment*. This includes software that is not privy to *Autocad* databases, either because it is unable to subscribe to them or none exist to access. While the computer industry has seen many proprietary protocols become *de facto* standards (for example, Hayes modem commands, HP *HPGL*™, Adobe *PostScript*™ and MS *Windows*™), this market-driven process tends to exclude users who don't have the necessary licenses, and can't guarantee that all their applications will be able to communicate effectively in any given setting. Yet even though they are proprietary, such arrangements point toward a better way of sharing spatial data.

Public and private institutions have to maintain corporate databases, which increasingly tend to have spatial components. Ratepayers and customers have addresses; departments and distributors have territories; citizens and consumers have postal codes and enumeration districts. Such facts can be normalized into relational fields and stored in standard tabular database management systems (DBMS). Locational facts and relationships used to spatially index records may be derived by GIS analysis, but it often makes sense to evaluate spatial queries in batches and store the results, rather than attempting to answer them interactively. The worth of this strategy depends on whether the convenience value of spatial indexes exceeds the costs of computing, storing and accessing them. Only an application can answer this for sure, and only for itself.

Making organized geographic facts available to client applications may best be handled by relational and object-oriented DBMSs. Many spatial entities and relationships can be described by querying objects, relations and views incorpo-

rating proximity, connectivity, set membership and other geometric and topological criteria. Much of this information is static, but updates and analyses can cause it to change. Only when new spatial relationships must be built need GIS muscles be flexed; most transactions involve queries that don't refer to coordinate data, and can be handled by aspatial DBMS engines.

Relying on DBMSs to handle GIS queries can work well, but isn't sufficient when users want to generalize or otherwise tailor map data or to explore map space (for example, finding features within buffered areas or querying proximal locations). Such pursuits require interactive access to graphic data and spatial operators, as well as to results returned from *ad hoc* queries. Still, many modeling applications (such as network routing/allocation problems and site suitability studies) can operate entirely with DBMS tabular data, needing graphic data only to display their results. This leads one to conclude that while access to graphics tends to be required only occasionally, the need is highly application-dependent, but when it exists, it may be strong.

If one's application is cartographic, tabular databases aren't much of a help. If not in Autodesk's orbit, one is probably stuck with databases built by the application's vendor; those that aren't have to be imported. There is some hope, however. As the result of military initiatives in "MC&G" (mapping, charting and geodesy), a new standard (a family of them, actually) for cartographic data communication has emerged, called DIGEST. Various implementations of it exist (Vector Product Format — VPF — is the most well-known), as well as one data product, the Digital Chart of the World (DCW), available on CD-ROM from USGS and other sources. DIGEST isn't a spatial data transfer standard; instead, it encapsulates complete, self-descriptive cartographic databases, ready for online prime time.

DIGEST can handle feature-oriented as well as layer-oriented map data. It can build points, lines and polygons into complex features and describe by degrees topological relations among them. Any data element can have any number of thematic attributes, but only those defined in its database's data dictionaries. There are places to put metadata as well as facts, organized into directories and files. Lots of files (with extensions like .ESR, .EDX, .FSI, .FAC, .END and .TXT), most of them tables. The reason there are so many file types is that many

of them are optional, only showing up when the datatypes they describe are defined: sometimes there are no point features or no text, or maybe topological relations aren't established. Omitting unneeded files is more efficient than storing empty fields, but makes applications work harder to understand a data model and parse its information.

Even a brief tour of DIGEST is well beyond the scope of this work. It is too complex and polymorphic to assimilate in any one sitting, and may be impossible to savor without the aid of software to collate and present facts and relations defined in any given realization (officially called a *profile*), such as DCW. That product is in fact distributed with software to browse, extract and view global map data, and would be virtually useless without it. But at this time, few GIS vendors have announced that they will support DIGEST, and none of them have promised to implement it as a native database mechanism. This should be neither surprising nor disappointing; until enough experience has been gained with DIGEST to validate its viability, it should not be naively embraced. More proven alternatives exist.

Standardized Spatial Query Language

Heterogeneous database environments are now very common, in large part thanks to the capacity to link them together via fourth-generation languages (4GL) for querying and updating databases. SQL and other declarative, non-procedural language interpreters permit data to be stored in private, proprietary formats while enabling access to it by any privileged user, locally or over a network. 4GL strategies are now widely used to integrate tabular data belonging to an enterprise, but users of geographic data have yet to be similarly blessed. Although many researchers and some vendors have developed spatial query languages (including extensions to SQL), there has been little progress in industry toward standardizing spatial query protocols.

It is not difficult to imagine the adoption of a set of operators that would enable spatial queries to be incorporated into an SQL *select* command. This basically means standardizing the semantics of a number of adjectives, such as *within, closer than, farther than, adjacent to* and *connected to* (there aren't a lot of them), and making provisions for parameters to qualify them. One would assume that if GIS vendors were strongly interested in interoperability of their databases, they

would have agreed on such protocols by now. But most vendors still lock up their customers' data in software safes from which data can be removed only by translation into interchange formats. They continue to purvey captive databases, although they increasingly recognize and accede to customer demands to integrate existing spatial data into the environments they engineer. Whatever data may exist in a customer's files, GIS vendors are delighted to install it in or link it to their databases. A flurry of custom conversion program creation usually ensues as data models are extended to handle new spatial constructs. This may well do the trick, but it begs the question, entails extra expense for customers, and yields *ad hoc* solutions that must be re-implemented over and over again.

One is understandably tempted to believe that GIS vendors are reluctant to make it too easy for users to access each other's databases using a client-server model. However, the reasons may not all be based on competitive advantage; while spatial primitives can certainly be accessed via standardized queries, it is genuinely difficult to communicate complete data models this way. How a layer-based GIS encodes geographic entities may be very different than how a feature-based one does. When the latitude users have in modeling data is added to this, it isn't hard to understand how SQL and proposed extensions to it might fail to express important aspects of spatial structure. While there are not a lot of spatial primitives that GISs tend to use, they are defined and glued together in systematic and arbitrary ways into objects, features, themes and layers. Unlike SQL, there is no public data language that can transparently access and relate spatial data and models held in proprietary GIS databases, nor is one likely to emerge for a number of years.

Application Programming Interfaces

Much what is and isn't going on in GIS data integration can be appreciated by considering technological forces in the PC software arena. Almost suddenly, after a decade or so of development, PC environments appear to be on the cutting edge of information processing technology. In many respects, the data-handling capabilities that personal computer users now or soon will have at their command rival those of workstation software, including GIS. Much of this power devolves from a rapid maturation of software architecture for handling complex documents.

There is an unmistakable trend in personal computer software engineering to provide increasingly open interfaces between applications in order to exchange "live" data. Many of these schemes, such as Microsoft's *OLE* (object linking environment) and Apple's *Amber* project, are object-oriented and describe protocols for integrating diverse datatypes such as word-processing documents, images, drawings, spreadsheets, video and sound. It is indicative that some Macintosh "works" programs integrate their component files via Apple's publish-and-subscribe system protocols, rather than via internal mechanisms such as calls, global variables and common databases. This style of architecture depends heavily on codification of an application programming interface (API) which standardizes protocols for accessing data from different applications and even for controlling them.

APIs aren't restricted to mediating interactions within and between compiled applications. With suitable utility software, the ability to generate, route and interpret inter-application requests can be placed directly in the hands of users. One way to achieve this is by empowering online queries to servers. SQL is an API to data which is usually thought of as a 4GL instead because it is uttered by users rather than software (although software may also formulate and issue SQL queries and directives). Although it is very useful, SQL by itself is not an adequate vehicle for communicating properties and applications of spatial data. To be effective, GIS data interchange must take place at both a lower architectural level and a higher conceptual level.

There is nothing mysterious about APIs. All software that has linked objects, functions or subroutines has at least one. Every call to a system or application library obeys the rules of some API, although the rules may change from system to system, application to application and language to language. So, while it takes a lot of work to develop a public, standard API, the task is not at all foreign to software engineers. Still, it does not seem to be easy to define an API satisfactory to all parties in the standardization process. Sadly, GIS vendors have been unable to agree on more than a few, and most of those are *de facto* industry standard APIs like Autodesk's DXF/DWG rather than collaborative initiatives.

The traditional role of APIs is to regularize access not to databases but to software. While the software may include database access functions, vendors of

GISs have been reluctant to publicize their own beyond a circle of trusted application developers. To fill this void, an increasing number of programming tools for accessing spatial data have appeared. They include software from Geodessy (Calgary, Alberta), MapInfo (Troy, NY), and TerraLogics (Lowell, MA). While each of these provides an impressive array of display, query and analysis functions, all depend upon proprietary, black-box database architectures that are not meant to be shared by GISs or other spatial information or mapping systems.

As one joke has it, the great thing about standards is there are so many to choose from. The same goes for APIs: interfaces for network data sharing; interfaces for database access; interfaces for graphic interchange; interfaces for GUIs... all are useful, but too many must be obeyed when designing applications. While incorporating APIs may yield more robust software, it is no easier to build and maintain than were pre-API standalone packages. Like other work, programming seems to expand to fill the time available.

Recently, a commercial paradigm surfaced that tries to deal with apibabble. Oracle Corporation (Redwood Shores, CA) announced software called Oracle *Glue*, intended to serve as a "universal API." Although *Glue* does not directly address specific issues of sharing spatial data, it is advertised as capable of handling all the low-level details of sharing databases among diverse applications in a networked environment of heterogeneous platforms. Oracle likes to call *Glue* "middleware" which is adaptable to a range of data services, portable to MS Windows, Macintosh, Unix and pen-based OSs, hardware-scalable, network-independent and capable of being accessed via 4GL commands as well as through 3GL programming interfaces. Even though it takes high ground in the API wars, this approach may fail to prevail. If all and sundry vendors are required to license it from Oracle, *Glue* may not stick to enough computing surfaces to be a useful adhesive. Even so, this approach has great merit, and points a possible way out of the abrasive nexus that holds GIS databases captive.

Conclusions

The art and science of spatial data handling faces a looming crisis in database access. Even though more than ten years have elapsed since GIS first walked out of the lab onto the street, and despite the number of systems that have since emerged, users and their databases tend to exist more as populations than as

communities, many islands of automation that only occasionally signal one other. A few simply reject direct questions, others can't answer them, and the rest prescribe what one can ask, and sometimes how to ask for it.

Most graphic data interchange standards are an evasion of the problem, not a solution. Think of how many have come and gone: SYMAP packages; CalComp plotfiles; Tektronix bytestreams; GKS metafiles; GINI and IGES; SIF and DXF; VPF and SDTS. Some had their day and deserved to die, others remain useful, but few offer online access to foreign data. When one compares this state of interoperability to that of tabular databases, one realizes how little has been accomplished and how much more data, currently inaccessible, may be at stake.

As this paper has tried to recount, there is clear evidence that spatial databases are on their way to opening their architectures. Although GIS and mapping system vendors haven't tended to be in the forefront of this movement, they should not be blamed for all the problems that need to be solved; spatial data is complex, heterogeneous, bulky and often application-specific. In addition, not every spatial database needs to be shared, nor is everything it contains of equal value to most users. But the pace at which digital spatial databases are accumulating is now so great that a real urgency exists for finding better ways to share them, and everyone who sells, builds and uses them must work together to broaden the pathways that connect them.

Formalizing Importance: Parameters for Settlement Selection from a Geographic Database*

Douglas M. Flewelling and Max J. Egenhofer

National Center for Geographic Information and Analysis
and
Department of Surveying Engineering
University of Maine
Boardman Hall
Orono, ME 04469-5711, U.S.A.
{dougf, max}@grouse.umesve.maine.edu

Abstract

This paper describes a model for selecting features from a geographic database to be displayed on a computer generated map display. An engineering approach is used to generate a set of geographic features similar to what would be chosen by a human, without attempting to replicate the human selection process. Selection is a process of choosing from a set of objects according to some ordering scheme. Humans have a highly developed ability to order sets of things. The model presented capitalizes on this ability by relying on user-defined ordering functions to produce an ordered set (*ranked list*) of geographic features. It is possible to process systematically the ranked list such that measurable qualities of the set such as subset relationships, pattern, dispersion, density, or importance are preserved. The resulting set of candidate features is placed on the map display using accepted generalization techniques.

Introduction

Geographic databases are usually too large to be presented in their entirety to a user—displays become too crowded to identify detail or pattern. Instead, a common approach is to equip a geographic database with a spatial query language, which allows a user to specify the data of interest. While this is appropriate when requesting specific data for which a user can provide some initial information such as attribute values or location ranges, it is very cumbersome when retrieving the data for a map-like presentation. To make a "good" selection, a user needs extensive knowledge about geography—what is important enough to be displayed—and cartography—how much can be displayed on a map. Therefore, methods are needed that select a representative set of data. Any such selection method has to preserve certain properties among the features in geographic space. For instance, taking geographic features randomly for a map would be unacceptable as the resulting map could not convey the characteristic geographic properties to the users.

A major factor contributing to the interest in the *intelligent selection of information* is the attempt to automate as much as possible the process of creating maps and cartographic

* This work was partially supported through the NCGIA by NSF grant No. SES-8810917. Additionally, Max Egenhofer's work is also supported by NSF grant No. IRI-9309230, a grant from Intergraph Corporation, and a University of Maine Summer Faculty Research Grant. Some of the ideas were refined while on a leave of absence at the Università di L'Aquila, Italy, partially supported by the Italian National Council of Research (CNR) under grant No. 92.01574.PF69.

products. Automation is desirable because it would shorten the time between getting from the raw material (the observations) to the end product (the map). For geographic information systems (GISs), where the resulting maps will be short-lived, this automation has a second perspective: map-like presentations have to be created quickly and often in an *ad-hoc* manner ("on the fly") the data stored. These data serve multiple purposes and are not tailored for any particular output representation. While maintaining the original purpose of the geographic inquiry, users frequently change "scale" through zooming, or select a different map extent through panning. Several factors influence such presentations: (1) parameters, such as the real estate and its resolution available to present the geographic information, constrain how much data to select; (2) parameters, such as the relative density in cartographic space, determine when placement conflicts arise; and (3) parameters that specify the *importance* of features and, therefore, govern what data to select. This paper deals with the latter issue.

Exploration of map displays by interactively zooming needs appropriate methods for selecting subsets of features to display from a geographic database. For the purpose of this paper, a geographic database is a model of reality. It contains information about geographic features that were collected without regard to the constraints of the maps in which the features will be displayed. This makes a geographic database different from a cartographic database, which represents a model of the document, the map, and includes all details about the rendering of the information in a particular cartographic space (Frank 1991). A geographic database is therefore likely to contain many more features than can be shown on any one map.

This paper contributes to a framework for engineering a software system that generates interactive map displays from a geographic database. Ideally, such map displays would match exactly the craftswork of a human cartographer in the content, geometry, and layout; however, human cartographers' products vary quite a bit, much like text covering the same topic written by different authors. No two human cartographers would produce exactly the same map (Monmonier 1991). Therefore, the yardstick for assessing the quality of a map display will be whether or not certain spatial properties have been preserved. The overall objective is to generate a satisfactory map within the limits of an interactive computer system. To meet the requirements of the system described above, the amount of data that is actually displayed has to be reduced. A selection from a geographic database must be made in such a way that it contains the features a human would have selected as candidates to be placed on a map. As a secondary constraint for an automated system, the selection mechanism should not require extensive processing to refine the selection as the features are placed into cartographic space.

The elements on which this paper focuses are *settlements*. Settlements are critical information on any map, because map readers frequently use them as references to orient themselves and because their distribution and density describe important geographic characteristics of a region. Settlements also have the advantage of being commonly represented as points on small and intermediate scale maps (Muller 1990). A point cannot spatially conflict with itself, therefore we are able to ignore most of issues of individual feature generalization and concentrate on sets of features.

The remainder of this paper first introduces the model used in this paper for settlement selection and then discusses settlements and their properties. This is followed by a discussion of methods for settlement ranking which presents a formal approach for define ranking functions. Finally, conclusions are presented with areas for further research.

A Model for Automating Settlement Selection

In their framework for generalization, McMaster and Shea (1992) place selection just prior to and outside of generalization, as do many other researchers in automated cartography (Kadmon 1972; Langran and Poiker 1986). McMaster and Shea go on to state that after an initial selection has been made, the set may have to be further reduced. So while "gross" selection is not cartographic generalization, "refinement" of the set of candidate features by removing features from the selection is cartographic generalization. These two different selections are characteristic of the difference between the operation on geographic space that produces the original subset, and the operation in cartographic space that acts on the subset due to the constraints of the map display.

In order to make a non-random selection of elements from a set, selection criteria have to be established according to which an element will be chosen over another. An *importance attribute* has been suggested as the primary attribute in the "Competition for Space" paradigm of map generalization (Mark 1990). Under this paradigm, all cartographic symbols compete for the limited real estate available on a map. Symbols that conflict because they occupy the same map area or are too close to each other to be distinguished, undergo an examination procedure in which the one with the highest importance or certainty factor is placed on the map, while the others are dropped. The same idea underlies most name placement techniques (Freeman and Ahn 1984; Langran and Poiker 1986; Mower 1986; Jones and Cook 1989), the most advanced sub-domain of automated map generalization.

Our model for transforming features from geographic into cartographic space builds on this notion of importance as the principal mechanism to preserve the semantics of geographic objects. In this transformation of geographic features from a geographic database to a map display in cartographic space, we identify three steps: (1) feature evaluation (or ranking), (2) feature selection, and (3) feature placement (Figure 1).

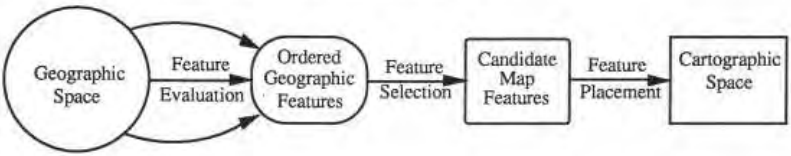


Figure 1: Transformation of features from geographic to cartographic space.

- *Feature evaluation* is the process of formally assessing the importance of each settlement in a geographic database. It generates for each settlement an *importance value* or *rank*. This procedure can be relatively simple, such as using the population of a settlement, or may involve several parameters such as a weighted balance between population, economic factors, connectivity, and availability of administrative facilities. After the evaluation is complete, the geographic semantics of the settlement have been encapsulated in the rank. The subsequent competition of map space is only based on the rank values of the settlements and their spatial locations.

  The process of ordering a set of settlements requires knowledge of the semantics of the attributes measured for the feature *and* an understanding of what information is required for the geographic problem being addressed. Defining the ranking for any set of settlements in any particular situation requires intelligence. Once the set is ordered, however, it is possible to process the set algorithmically to generate results that closely resemble selections made by human cartographers. Answering any geographic question

requires structuring of the available data with an ordering function. Ideally the ordering function should produce a single unique value for every member of the set.

- *Feature selection* extracts candidate settlements from the ranked list. There are several possible methods of selection depending on the properties that the user wants to preserve, similar to the manner in which cartographic projects are designed to preserve one or more spatial characteristics important to a map (Tobler 1979). Several of these methods have been discussed before (Flewelling 1993).

  In this model, it is assumed that not all of the settlements in the database can physically fit on the map. The process of selecting from the ordered set is on the surface simple, but an examination of published maps indicates that other criteria may be being used. There are several other factors, such as visual density, pattern, or dispersion (Unwin 1981) that geographers may wish to preserve when choosing from a set of features. Ideally, these factors should be preserved regardless of the scale of the display.

- Finally, *feature placement* introduces candidate settlements in rank order and resolves spatial conflicts in the cartographic space. It is assumed that the settlements will be represented as point features with labels. As such there are relatively few generalization operations that can be used (McMaster and Shea 1992). If settlements are placed in rank order and there is spatial conflict on the map, a settlement can be either be displaced or not placed. This is similar to name placement schemes, which usually operate as iterative processes. Successive attempts are made to find a suitable label position; only as a last resort is the feature not placed.

The particular advantage of this three-step model is that once a set of features is ranked, it is possible to produce different scale maps without addressing geographic issues for each new map display. The cartographic rules can take into consideration such criteria as screen size, display scale, display resolution, color depth, printed scale, and output media quality. None of these issues affect the underlying semantics of the features, rather they address the requirements for a legible map.

Properties of Settlement Attributes

Settlements have both spatial and descriptive characteristics, both of which may be used to determine different aspects of importance. In our model of settlement selection, the descriptive information related to each settlement in the set, such as population or number of hospitals, governs the ranking of settlements. Spatial attributes such as location and shape of a single settlement contribute to generalization at later stages of map construction.

While an exhaustive list of the descriptive parameters that can be measured for a settlement is impractical, it is possible to consider the classes of values that might be collected and determine their utility in generating an ordering for the settlements. At the simplest level, it is possible to record the presence or absence of a characteristic. For example, a city is or is not a capital. This produces a dichotomy from which it is possible to make some judgment about a set of features, but dichotomies do not have any inherent ordering. The ordering imposed on the dichotomy is simply a decision that either the presence or absence of the attribute is more important. After this decision has been made no further distinction exists among the elements of both groups.

Binary data are difficult to use with a scale-based selection threshold. There will be a range of scales where there are too few and another where there are too many features on the map. As soon as there are too many features, the only option available is to no longer show any of the feature in the set. For example, at a scale of 1:1,000,000 it is possible to show

every county seat in the United States, however at 1:10,000,000 it is no longer possible to show them all. Without additional criteria it is impossible to select from among the capitals to determine which county seats will be removed from the set. Such a threshold may have its uses in defining acceptable scale ranges for particular classes of features.

In order to be rankable, settlement attributes must have particular properties. The most common categorization of attribute data is the classical distinction of nominal, ordinal, interval, and ratio (Stevens 1946).

Nominal

Nominal data allows for the comparison of "equality" of two values, but not more. Settlement names are an example of nominal data. The name of a settlement is simply a tag to identify the place as being different from some other place. While names are important properties to identify, this piece of information is of little use in generating a complete ordering over a set of settlements. It is possible to classify settlements in terms of the relative sizes with terms such as metropolis, city, town, village or hamlet. This partial ordering of nominal values makes it possible to perform a limited amount of selection by choosing the classes of settlement to be displayed. For instance, Christaller's (1966) central place theory provides a means to classify settlements into a partially ordered set. The geographic space is divided into regions where a particular settlement is the economic center for a number of subordinate settlements. A hierarchy is developed where there are first-order settlements serving second order settlement which in turn serve third-order settlements, and so on. The result is partially ordered because it is impossible to determine whether a second-order settlement in region A is greater in rank than a second-order settlement in Region B. In this particular case there is a much lower chance that this will cause a problems with spatial conflicts because the entire space is hierarchically partitioned. This is also the case with most administrative regions, but not the case with features routes. While routes might be classified into a partially ordered set (e.g. Interstates, U.S. Routes), they often share geographic space, or due to the realities of geographic terrain, are in spatial conflict in the cartographic space.

Ordinal Data

Ordinal data establish a sequence of the elements, S_i, through an order relation, r, which is reflexive, antisymmetric, and transitive.

$$S_i \, r \, S_i \tag{1a}$$

$$i \neq j: \quad S_i \, r \, S_j \;\Rightarrow\; \neg(S_j \, r \, S_i) \tag{1b}$$

$$(S_i \, r \, S_j) \wedge (S_j \, r \, S_k) \;\Rightarrow\; S_i \, r \, S_k \tag{1c}$$

While one can determine whether one objects A comes before another object B in this sequence, it is unknown *how much* there is between A and B; however, this is not a problem in ranked lists where all that is necessary to be known is the sequence of the elements. It has been assumed in this model that the geographic semantics are encapsulated in the order.

Interval and Ratio Data

For the purposes of generating an ordered set of settlements there are no differences between interval and ratio data. Both kinds of data are true metrics since the intervals

between values are constant. The only difference being whether the zero point is set arbitrarily (interval) or not (ratio). In either case it is quite simple to determine whether the population, for example, of settlement A is greater than settlement B. For the purposes of selecting from a set of features the greater flexibility of interval and ratio data is not required. The power of these kinds of data is in the much more complex ways in which they may be combined and analyzed to produce an ordering function. For instance, Kadmon's (1972) work defines a complex set of weights and values that he used to order a set of settlements in Israel. After the set was ordered a decision was made about how many settlements where to appear on the map and the top n settlements were selected.

Ranking Settlements

Ranking a set of settlements is a transformation from a relatively information rich environment into a simple and constrained environment. In the geographic database there can be any number of parameters measured for an individual settlement. Choosing the correct function that manipulates these parameters requires cognition and understanding of the problem being addressed. For instance, in his view of the world Mark Jefferson (1917) concluded that population alone was all that was necessary to separate the important settlements from the trivial. Other techniques for ordering settlements such as those used in central place theory (Christaller 1966; Lösch 1954; von Thünen 1966) require examination of multiple variables in more complex ways.

Whatever the actual ordering function used, it is possible to state that given a set of settlements S and an ordering function o which uses the parameter values recorded for a settlement, there is a transformation function f that can produce a ranked list of settlements R:

$$f(S, o(\ldots)) \rightarrow R \tag{2}$$

such that both sets have the same cardinality (#), i.e., $\#(S) = \#(R)$.

When one considers the types of information gathered about settlements, such as population, it is clear that it is impracticable to require unique ranks from this function. The probability of two settlements having equal parameter values is almost certain in a geographic database of even moderate size (Figure 2). For instance, two settlements can have the same population. Although two settlements cannot share the same location, there are cases (such as Rome and Vatican City) where two settlements are co-located when they are represented as points. It is also possible for settlements to be in the same location, but at different times. Therefore, it is necessary to develop a means for handling *complex ranks* in a partially ordered list, where a particular rank value may be attached to more than one element of the ranked list. In such cases, a decision must be made to use either selection by classification or to permit first-in resolution of conflicts.

It is possible to combine different rankings with different weights to produce a new ranking that considers multiple parameters. This approximates a more complex ordering function in f. For example:

S = Set of all Settlements
R_i = Ranked List of Settlements
W_i = Weighting Factor

$f(S, Population\ ()) \rightarrow R_{pop}$
$f(S, Employment\ ()) \rightarrow R_{emp}$
rank $(W_1 \cdot R_{pop} + W_2 \cdot R_{emp}) = f(S, Pop\_Emp\ ())$

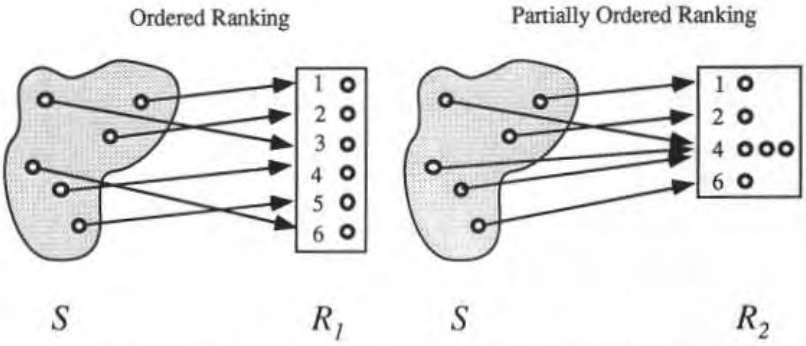


Figure 2: Transformation from an unordered set onto (a) an ordered and (b) a partially ordered ranked list.

Within the constraints of the engineering approach presented here, it may be possible to generate settlement selections with weighted rankings that are very similar to rankings created by a more complex ordering function (Figure 3). The practice is well accepted in applied and academic geography (Unwin 1981; Boyer and Savageau 1989). By providing a mechanism for storing "authorized" rankings prepared by application experts, domain specific knowledge could be encapsulated for use by non-experts.

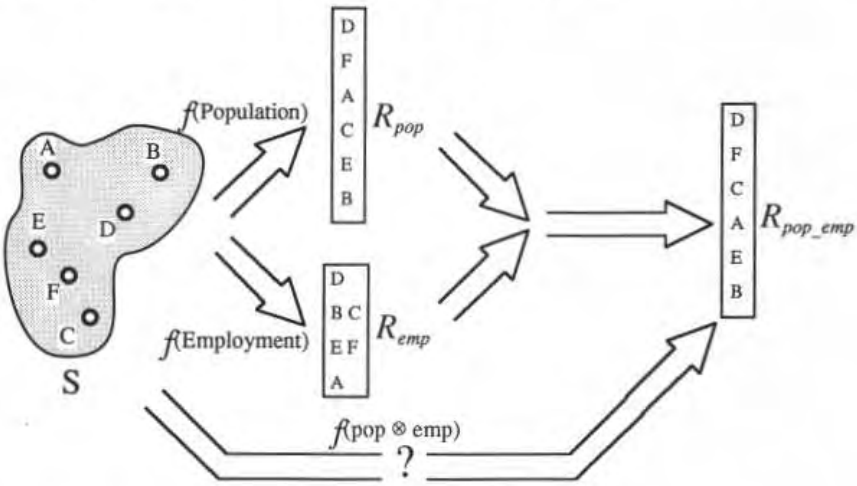


Figure 3: Combining simple rankings to approximate complex rankings.

Conclusions and Future Work

This paper has presented a model for automated map construction for an interactive GIS. The system is constrained by a requirement for frequent and rapid reconstruction of maps as scale and spatial extent change. The maps are short lived, so a reasonable approximation of the feature sets that would have chosen by a human is acceptable, as long as the most important features are shown. Settlements were chosen as the feature set to test this model.

It was shown that selecting a subset of settlements for a particular map relies on there being an order placed on the set of settlements. A random selection from the set does not preserve the geographic relationships desired for useful analysis. A framework for ordering settlements by evaluating a settlement's non-spatial parameters and ranking those evaluations has been described. Ideally the result should be fully ordered, but in practice this is difficult to achieve. Therefore, methods for processing the partially ordered list are necessary.

In this paper, no strategies for processing the ranked list were discussed in detail. Mark's (1990) "Competition for Space" paradigm suggests that only a rank order processing of the list is necessary. Where there is spatial conflict, either a "compromising" generalization method (e.g. displacement) would be used or the feature would not be placed. Whether or not this will produce an acceptable map has yet to be determined. Other selection strategies that address spatial parameters may be required to generate appropriate sets. This research is currently being conducted by the authors.

The degree to which a weighted ranking approach approximates a more complex ranking function must be investigated. If the weighted ranking approach results in selections that do not vary significantly from those in published maps then the issue becomes a matter of how rapid the system can respond to user actions. The most responsive system is one that can produce enough information about the "real" geographic world on a map for the user to make a decision that is valid in the real world.

Acknowledgments

Discussions with Andrew Frank, David Mark, and Scott Freundschuh provided useful input. Thanks also to Kathleen Hornsby who helped with the preparation of the manuscript.

References

Boyer, R. and D. Savageau (1989) *Places Rated Almanac*. New York, Prentice Hall Travel.

Christaller, W. (1966) *The Central Places of Southern Germany*, Englewood Cliffs, NJ: Prentice Hall.

Flewelling, D.M. (1993) Can Cartographic Operations Be Isolated from Geographic Space? Paper presented at the *1992 Annual Convention of the Association of American Geographers*, Atlanta, GA.

Frank, A.U. (1992) Design of Cartographic Databases. in: J.-C. Muller (ed.), *Advances in Cartography*, pp. 15-44, London: Elsevier.

Freeman, H. and J. Ahn (1984) AUTONAP—an expert system for automatic name placement. In: D. Marble (ed.), *International Symposium on Spatial Data Handling*, pp. 544-569, Zurich, Switzerland.

Jefferson, M. (1917) Some Considerations on the Geographical provinces of the United States. *Annals of the Association of American Geographers* 7: 3-15.

Jones, C.B. and A. Cook (1989) Rule-Based Cartographic Name Placement with Prolog. In: E. Anderson (Ed.), *Autocarto 9*, pp. 231-240, Baltimore, MD.

Kadmon, N. (1972) Automated Selection of Settlements in Map Generalization. *The Cartographic Journal* 9(1): 93-98.

Langran, G.E. and T.K. Poiker (1986) Integration of Name Selection and Name Placement. In: D.F. Marble (Ed.), *Second International Symposium on Spatial Data Handling*, pp. 50-64, Seattle, WA.

Lösch, A. (1954) *The Economics of Location,* New Haven, CT: Yale University Press.

Mark, D.M. (1990) Competition for Map Space as a Paradigm for Automated Map Design. In: *GIS/LIS '90*, pp. 97-106. Anaheim, CA.

McMaster, R.B. and K.S. Shea (1992) *Generalization in Digital Cartography* Washington, DC: Association of American Geographers.

Monmonier, M.S. (1991) *How to Lie with Maps.* Chicago: University of Chicago Press.

Mower, J.E. (1986) Name Placement of Point Features Through Constraint Propagation. In: D.F. Marble (Ed.), *Second International Symposium on Spatial Data Handling*, pp. 65-73, Seattle, WA.

Muller, J.C. (1990) Rule Based Generalization: Potentials and Impediments. In: K. Brassel and H. Kishimoto (Eds.), *Fourth International Symposium on Spatial Data Handling*, pp. 317-334, Zurich, Switzerland.

Stevens, S.S. (1946) On the Theory of Scales of Measurement. *Science Magazine*, 103(2684): 677-680.

Tobler, W.R. (1979) A Transformational View of Cartography. *The American Cartographer* 6(2): 101-106.

Töpfer, F. and W. Pillewizer (1966) The Principles of Selection. *The Cartographic Journal* 3(1): 10-16.

Unwin, D. (1981) *Introductory Spatial Analysis*, New York: Methuen and Co.

von Thünen, J. H. (1966) *Isolated State*, London: Pergamon Press Ltd.

CALCULATOR: A GIS CONTROL PANEL FOR EXTENT, SCALE AND SIZE MANIPULATION

John H. Ganter, GIS Specialist
GRAM, Inc.
8500 Menaul Blvd. NE, Suite B370
Albuquerque, New Mexico USA 87112
Tel: 505-299-1282
Internet: jganter@ttd.sandia.gov

Todd E. Crane, GIS Specialist
Consolidated Rail Corporation
2001 Market Street
Philadelphia, Pennsylvania 19101 USA
Tel: 215-209-5681

ABSTRACT

Most GIS interfaces handle extent (dimensions on the ground), scale (e.g., feet per inch), and size (dimensions on screen or media) by solving the equation [Extent / Scale = Size] in a forward manner. While easily understood, this approach requires users to perform iterative experiments or scratchpad calculations if constraints on one or more of the variables are present. When large maps (e.g., 3 feet and longer produced on plotters) are involved, this problem contributes to the difficulty of layout on relatively small workstation screens.

Using the interface tools in Arc/Info Macro Language (AML), we developed a prototype control panel (Calculator) that allows users to manipulate the equation parameters, dependence of variables, and direction of solution. The Calculator is a *visual formalism* similar to a spreadsheet. Results are immediately visible to the user both as numerical values on Calculator and in the map layout. The Calculator was then linked to other panels that control detailed aspects of layout (e.g. spacing between legends and titles, etc.) so that the resulting sums of dimensions can be controlled.

We describe the simple algorithms and complex decision tree of the underlying code. This calculator analog appears to be a useful supplement to direct layout manipulation for experienced GIS users.

INTRODUCTION

Large maps are a popular and important product of GIS. Maps up to several feet long, in large scales such as 1000 feet per inch[1], are often requested for facilities design, use at construction sites, earth science and environmental fieldwork. Public displays, presentations, posters, etc. can require even larger dimensions. In these sizes, the time, expense, and wasted paper and inks of bad plots make a single run highly desirable. (This is especially true if the plot is done offline by a service bureau.)

Direct manipulation of graphic layout is used in most what-you-see-is-what-you-get (WYSIWYG) graphics packages. The user moves a cursor (by operating a mouse, trackball, etc.) to select, move and resize graphic objects (Mark and Gould 1991). This approach gives the user a great deal of control and

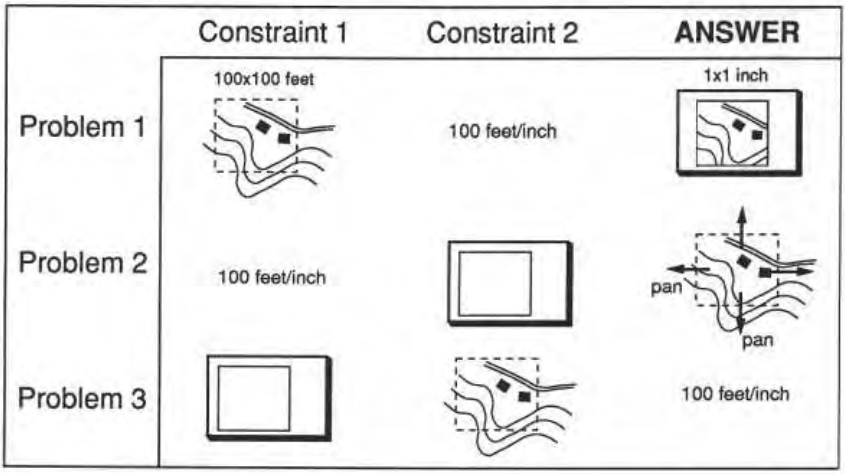


Figure 1: Common Extent/Scale/Size problems in map layout

flexibility, as they perform actions and receive instant feedback on the consequences of those actions.

For large maps, direct manipulation is difficult to utilize because of the relatively small dimensions of even 'large' workstation displays. Also, direct manipulation is not always optimal for fast production of many maps with similar, but slightly different, dimensions.

As part of the General GIS Interface (GGI) (Crane 1993; Ganter 1993a and b), we decided to build a control panel that would allow the user to calculate, manipulate, and constrain the variables of Extent (dimensions on the ground), Scale (e.g., feet per inch), and Size (dimensions on screen or media) in an interactive and iterative manner. We thought that such an interface might improve the user's conceptual model of the subtle relationships between these variables and their practical consequences in map layout. It would also free time for other tasks, including the more aesthetic aspects of map design.

EXAMPLE PROBLEMS

The Calculator panel assists the GIS user in solving three common problems (Figure 1):

(1) Given an Extent on the ground, and a definite Scale, what is the Size of the resulting map? Example: several buildings must be shown, the Scale must match engineering maps; can output be laser printed or must it be plotted?

(2) Given a fixed Scale and map Size, what Extent on the ground can be shown? Example: The Scale must match 1:24,000 topographic quadrangles and the map must fit on letter-size paper. What size zone can be shown

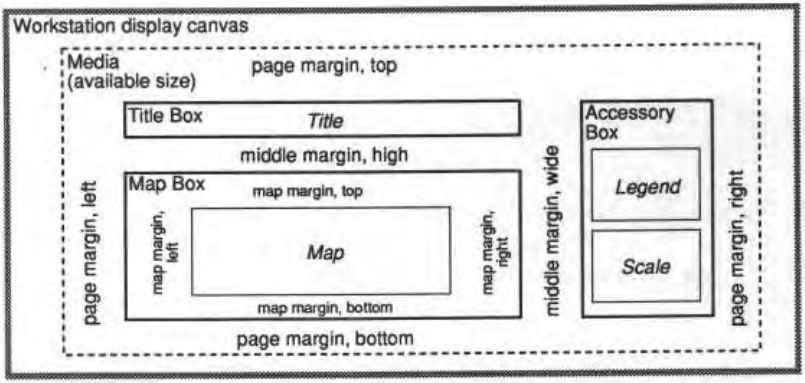


Figure 2: Map layout template

around the feature of interest?

(3) Given a fixed map Size and Extent, what is the Scale? Example: a facility must be completely shown on letter-size paper; what is the Scale? If a decimal results, the user can round it up or down to a large round number as desired, and test the result in (1) or (2) above. For instance, a user would probably round 96.7 feet per inch to 100 feet per inch.

A manual approach to solving these problems could involve many steps, especially if the layout includes legends and other accessories. For instance, in a variation of Problem 3, the user might have roughly 6 inches available in width, 4 inches in height, and an extent of 400 feet wide by 500 feet high. Scratchpad calculations would follow:

$$\begin{aligned}
Scale_{width} &= 400 / 6 = 66.66 \text{ feet per inch} \\
Scale_{height} &= 500 / 4 = 125 \text{ feet per inch} \\
Size_{width} &= 400 / 70 = 5.7 \text{ inches} \\
Size_{height} &= 500 / 70 = 7.1 \text{ inches} \\
Size_{width} &= 400 / 125 = 3.2 \text{ inches} \\
Size_{height} &= 500 / 125 = 4.0 \text{ inches}
\end{aligned}$$

By doing a sequence of such calculations, and then entering the results into the GIS and observing the appearance of the display, the user can eventually come up with a satisfactory layout. The user might also set up forms of the equation in a spreadsheet to aid this experimentation.

We decided to extend the spreadsheet idea into a kind of specialized 'Calculator' linked to the graphic display in two ways. First, the Calculator would control the display; typing in a new extent would change that extent in the display. Second, manipulating the graphic display (e.g., by zooming in) would change the Calculator values. The user could also *lock* certain values in the Calculator that they wished to preserve, thus forcing the Calculator to change the way it solves the equation.

The Calculator can be considered a *visual formalism*. Visual formalisms are defined as "diagrammatic displays with well-defined semantics for expressing relations" (Johnson, et al., 1993). Visual formalisms in user interfaces are

control panels, expanding trees, schematics, etc. that use widgets (buttons, readouts, sliders, dials) in arrangements that conform to user's knowledge and expectations. Calculator benefits from three general qualities of visual formalisms (Johnson, et al., 1993):

- VISUAL COGNITION: humans have both innate and acquired skills in recognizing and inferring abstract structure from spatial relationships such as proximity, linearity and alignment;

- FAMILIARITY: they are part of everyday life, so people become adept at recognizing similarities and experimenting to figure out how they operate;

- MANIPULABITY: they are dynamic, changing and reacting to both reflect and constrain user actions.

These general characteristics suggested that we could design a panel with a high density of changing data, and that users would be able to discern basic structure and function from their past experiences with everyday objects like calculators, tables, and spreadsheets. We hoped that the Calculator would aid the user in learning, enhancing, and applying a conceptual model of the extent/scale/size equation. The Calculator would provide feedback to either solve a layout problem immediately, or encourage experimentation that would reveal one or more design solutions.

DESIGN AND LAYOUT OF PANELS

As mentioned, the Extent/Scale/Size calculation outlined above must be extended somewhat to handle more complete layouts where the map is surrounded with titles and legends. We control these cartographic devices as 'Boxes' in a standardized template (Figure 2), the dimensions of which the user can modify. The user can experiment with various Box sizes and the margins between them to alter the overall appearance and dimensions of the

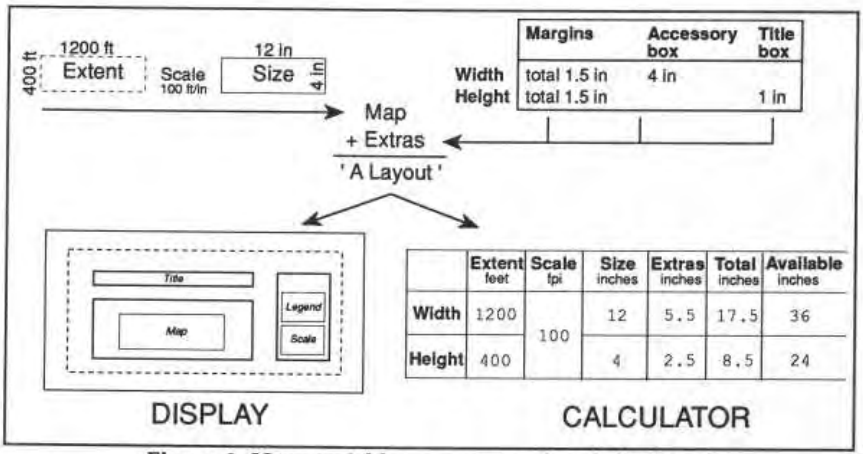


Figure 3: How variables are summed and displayed

map. We sum all of the space consumed by these *Extras* (margins and Boxes) in the horizontal and vertical dimensions, and display these sums in the Calculator (Figure 3).

The user is thus assisted in calculating a map Size (Figure 3, top left) that is combined with Extras to produce results (a total Size) on both the graphic display (Figure 3, left) and the Calculator panel (Figure 3, right). The user can see how the Extras contribute to the Total layout size and how this compares to the Available dimensions of the currently selected output device.

Our overall interface uses the analogy of a camera which is directed by the user at places of interest. The Calculator (Figure 4, bottom) can be thought of as controlling the magnification of the camera lens. The user clicks 'Lock' buttons to choose which variables are the 'constraints' (see Figure 1). For

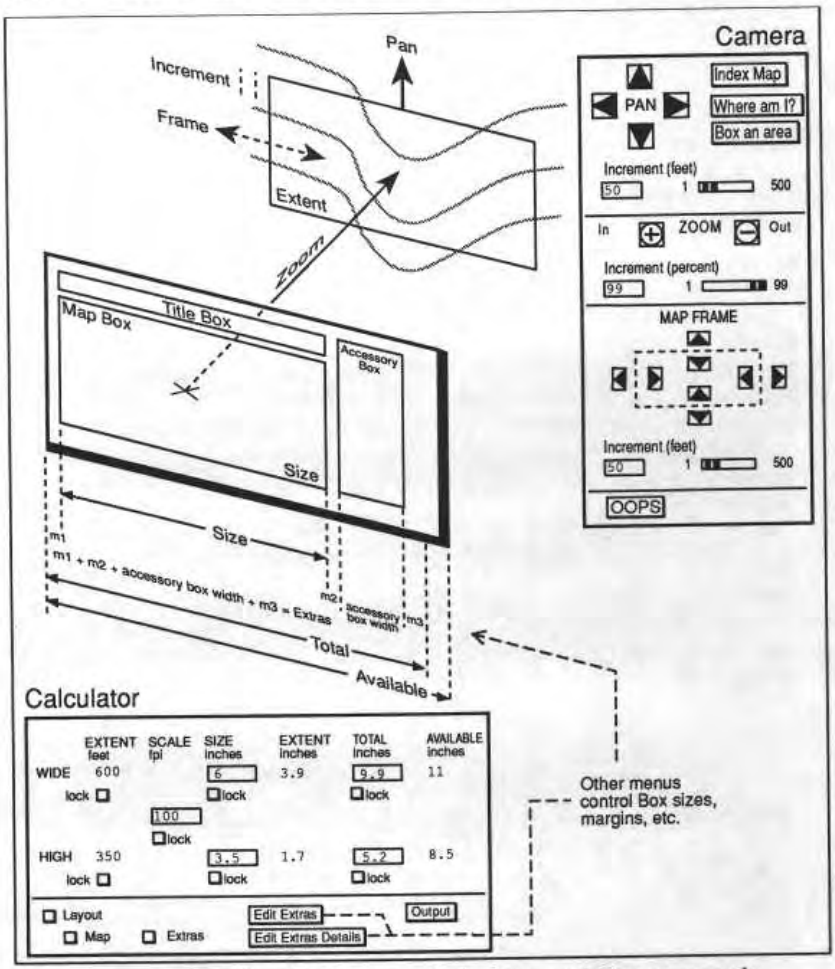


Figure 4: Relationship between Calculator and Camera panels and the map layout

example, to solve a problem of fixed Extent and fixed Scale, three Locks (Extent height, Extent width, Scale) would be depressed and the Calculator would then report the Size.

Continuing with the camera analogy, another adjacent panel controls pan, zoom, and framing (Figure 4, right). Panning merely moves the camera lens: it has no effect on the Calculator. Zooming is closely linked to the Calculator and can affect Extent, Scale or Size depending on the status of the Locks. For example, if the Extent is locked then the user is not permitted to Frame.

(Each Camera control has sliders that control the increment: the amount, either in units or percentage of the screen, of movement for each button press. The panel also contains several loosely related buttons. Index Map shows the initial 'start-up' view; the whole Extent of the present database. Where am I? temporarily zooms outward and draws a box around the current Extent. Box an Area is a form of zoom where the user draws a box, using mouse and cursor, around the area of interest. Oops is an undo/redo feature that cancels/uncancels the last action.)

At the bottom of the Calculator are two buttons, Edit Extras and Edit Extras Details, that control the detailed dimensions that combine into the Extras totals. Edit Extras is more frequently used and brings up a panel for controlling margins and the sizes of the Title Box and Accessories Box. Edit Extras Details is less commonly used, and allows some control over the internal characteristics (e.g., fonts, line weights, colors) of Boxes. More control is expected in future versions, so that the user can modify the parameters of the algorithms that perform a 'best fit' of the legend, scale, etc. in the Boxes (Ganter 1992).

Another button on the Calculator, Output, brings up a panel (not shown) where output device and media dimensions and orientation are chosen. This panel influences the Available inches readout that is displayed on the Calculator.

At the bottom left of the Calculator panel are three buttons that control the manner in which the map and extras are shown on the display canvas. When Layout is toggled off, the current Extent fills the available screen space. This is ideal for simply browsing data, typically by using the Camera control panel. When Layout is toggled on, the display canvas reflects current Size. For example, if the Size is 30 inches, the user will see 30 inches drawn to scale as a miniature (and unreadable) map. In this Layout view, the user can choose to see the map, the Extras, or both. This feature is useful during the iterative process of designing a layout, since it allows the user to concentrate on one or the other without waiting for the entire screen to redraw.

EXAMPLE OF USE

The user can interact with the Calculator both by typing values into the readout spaces, and manipulating the Camera panel. As a result, the Calculator readouts are very dynamic but can be constrained by the Locks. The user is notified of invalid operations (e.g., attempting to Frame when the Extent is

locked), and the system suggests appropriate actions though pop-up message boxes.

Figure Sequence:

KEY	
⌜	button up (off)
⌟	button down (on)

5 The user views a number of polygonal features. The Extent, Scale and Size are visible on the Calculator.

6 Using the Box an area button on the Camera panel, the user has zoomed in to view two polygons. The user has also changed the Scale to 90 feet per inch, and locked both this scale and the Extent so that they will not change. The new Size is displayed, and the Extras, while not shown, are calculated.

The user then decides to set the Total size to 10.5 inches, in order to more completely fill the Available media. They enter and lock a Total width of 10.5 inches. This additional space must now be dealt with.

7 The Allocate menu appears automatically. The user is told that there is 0.5 inches available. This space can be used in the Accessories box, or various margins.

8 The user increases the Accessories box width from 2.4 inches to 3.9 inches, thus consuming the extra space. The new Extras and Total inches are displayed. The Layout button is toggled, so that the complete Layout (map and extras) is now drawn.

THE CALCULATOR EVENT LOOP

AML menus allow the programmer to construct an interface with a few high-level function calls under which the details of the low-level X11 graphical operations are hidden. These menus are event-driven. Their normal state is 'at rest' displaying the status of global variables (e.g., an Extent value or Lock status), while waiting for an 'event:' a mouse click or keyboard entry.

When an event occurs on Calculator, a series of tests and calculations are performed that check the validity of the event, the context in which it occurs, and the effects on other variables. The event, if valid, disturbs an equilibrium in the equation [Extent / Scale = Size]. The equilibrium is then restored by propagating changes to all the affected variables and redrawing the screen.

The general steps in restoring this equilibrium are as follows:

(1) The menus display the current values of variables and locks.

(2) The user presses a button on the Camera panel (e.g., Zoom), or types a new value in Calculator.

(3) Step 2 produces an instruction to process specific variables. A program takes this instruction, checks it for validity (e.g., Zoom is not possible if the Scale or Extent are Locked). The program then requests information on the

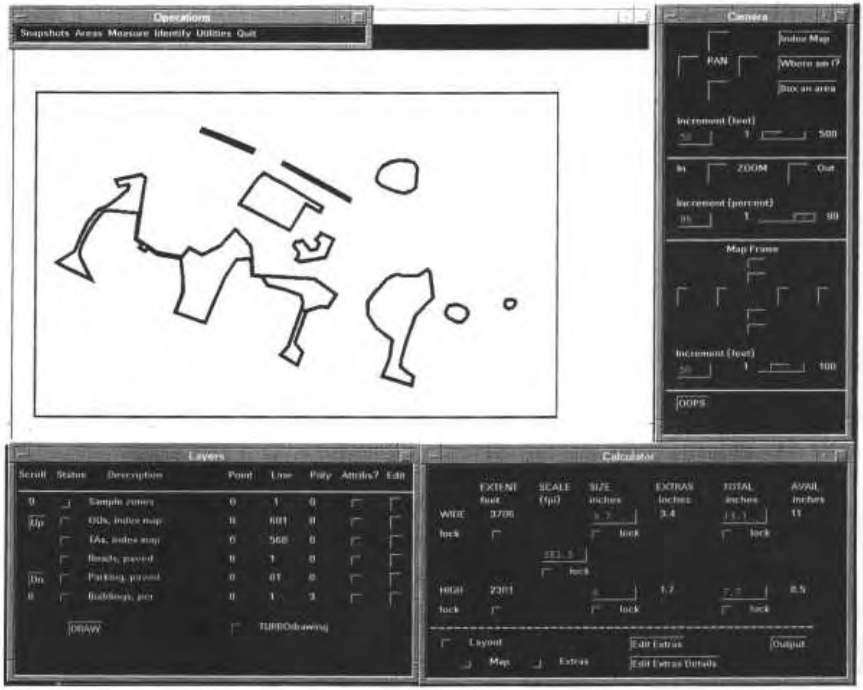


Figure 5

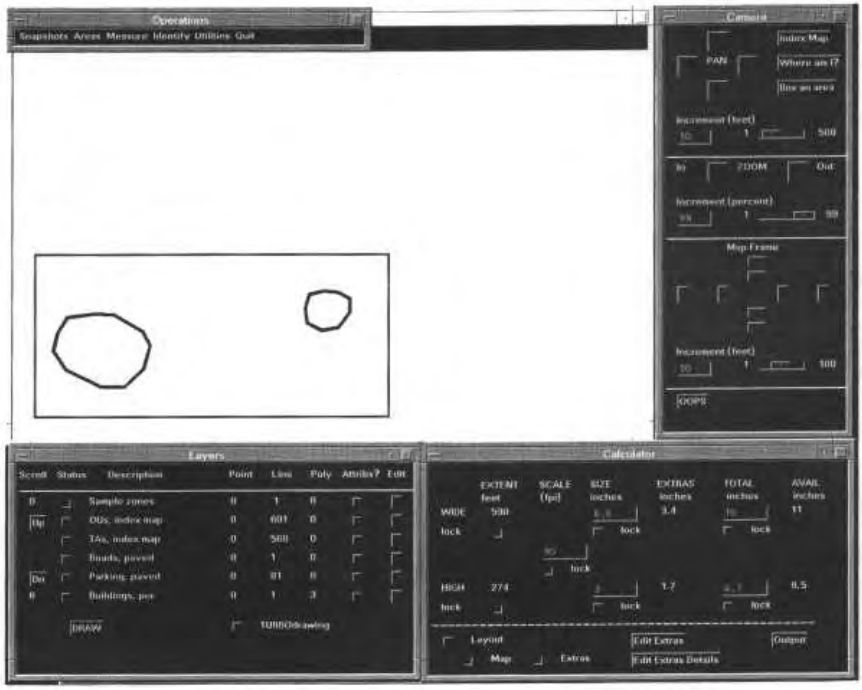


Figure 6

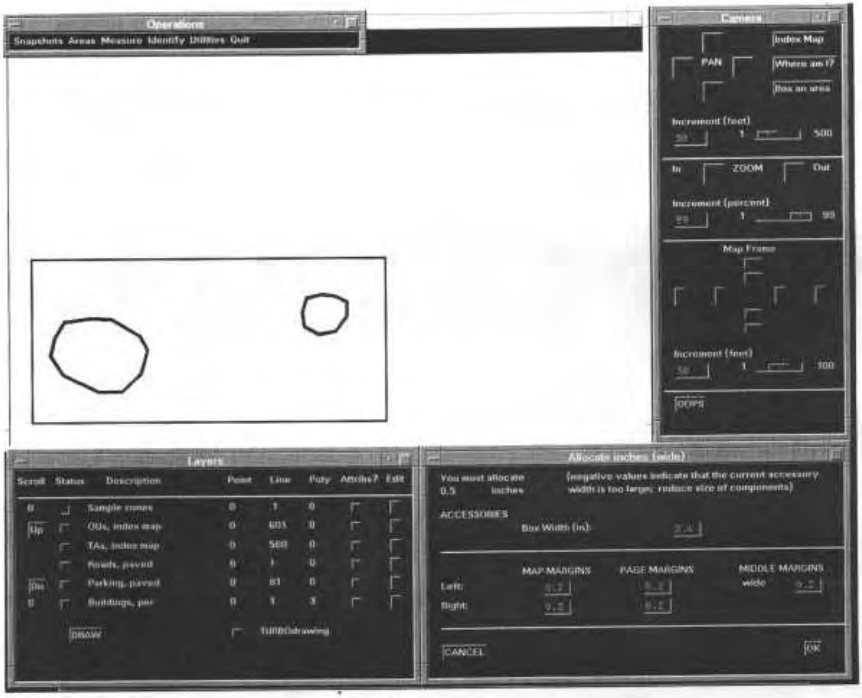


Figure 7

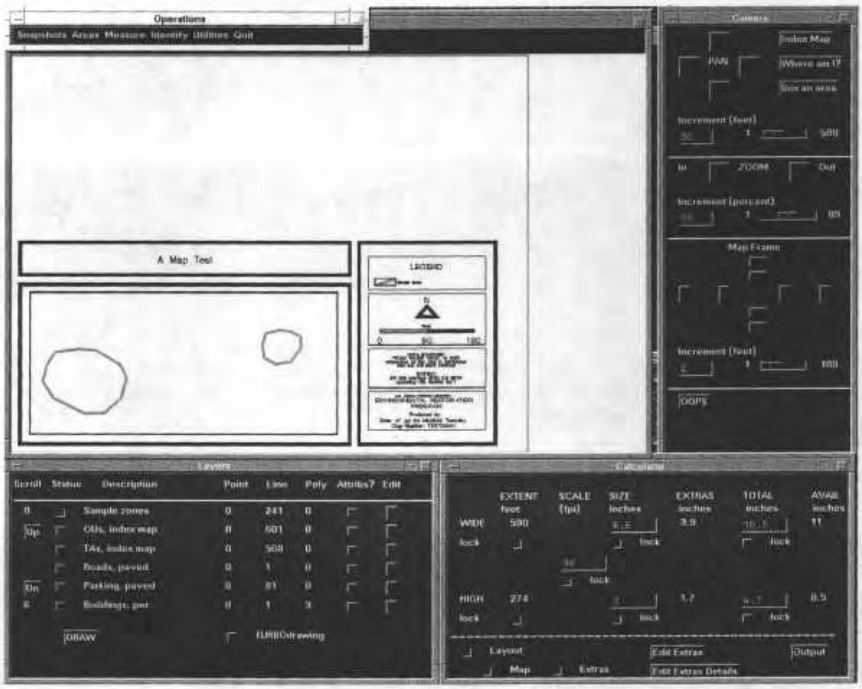


Figure 8

form of the equation to solve and the calculations that are necessary.

(4) A subroutine concatenates the original instruction with a binary string that represents the status of the seven Locks. This complex state is matched to an entry in a lookup table to determine which calculations to perform. There are over 100 distinct combinations of instructions and Locks in the lookup table.

(5) The calculations are performed, and the new values are propagated to all affected variables. Equilibrium is restored, and Calculator awaits the next event.

CONCLUSIONS

Calculator seems to support the visual formalism concepts of using innate visual cognition, familiarity, and manipulability. Our test users were able to grasp the operation of Calculator, and felt that it behaved like a somewhat unusual spreadsheet. In learning to use Calculator, it was useful to let it simply 'freewheel' with all Locks turned off while various Camera operations were carried out. By observing the behavior of Calculator while doing simple pans, zooms and frames, users developed a grasp of the variables it was showing and how they could influence these directly. Calculator also proved to be oddly complex at times, and it is possible that simplifications, additions, or even a completely different approach would be more illuminating for the user.

At present, Calculator requires the Scale to be in feet per inch. This is a useful and intuitive form for large scale maps, but the user should have the option of using ratio scales (e.g. 1:24000), and metric units.

The use of templates for layout, and the presently limited control of those templates, may seem rigid but it does support the rapid production of generic maps. This can free time for pursuit of more creative options using other tools, e.g. Arc/Info Map Composer, or drawing packages. The Calculator is a key to this intermediate *draft map generator* between GIS data and final cartographic production.

The menu functions are a significant addition to AML that makes rapid prototyping of concepts like the Calculator possible without low-level programming of the X11 graphical user interface. However, at the current version there is limited control of panel layout and graphical elements. For instance, it is quite difficult to precisely align widgets, and to draw lines and separators. It would also be useful if the readouts could be 'flashed' to alert the user to a new result or required entry.

ACKNOWLEDGEMENTS

Early work on this topic was done while the authors were affiliated with the Environmental Restoration Program at Los Alamos National Laboratory (LANL). LANL is operated by the University of California for the US Department of Energy. This publication has not been formally reviewed, and

all opinions expressed are those of the authors. Mention of commercial products does not constitute endorsement by any organization. Data shown are fictionalized for the purposes of illustration.

The support of Sandra Wagner, Paul Aamodt, Lars Soholt, and program manager Robert Vocke is gratefully acknowledged. GRAM, Inc. supported portions of report preparation, for which we thank Krishan Wahi.

NOTE

1. By convention, the scale of a map is based on the ratio of map distance / ground distance. A 1:100000 map is thus "smaller scale" than a 1:24000 map, but it shows a larger area.

REFERENCES

Crane, T. 1993, *A Graphical User Interface for Map Production within the Environmental Restoration Program at Los Alamos National Laboratory*: Unpublished MA project, State University of New York at Buffalo, 72 pp. [LA-UR number pending.]

Ganter, J. 1992, Software-Assisted Layout of Arc/Info Maps (abstract): *Los Alamos National Laboratory unpublished report LA-UR 92-2875.*

Ganter, J. 1993a, Metadata Management in an Environmental GIS for Multidisciplinary Users: *GIS/LIS '93*, Minneapolis MN, 2-4 November 1993, in press.

Ganter, J. 1993b, Design and Development of a Geographic Information System Interface for Environmental Restoration Data Access and Map Production: *Environmental Remediation '93 Conference*, Augusta GA, 24-28 Oct. 1993, in press.

Johnson, J., B. Nardi, C. Zarmer, J. Miller. 1993, ACE: Building Interactive Graphical Applications: *Communications of the ACM*, Vol 36:4, April, pp. 41-55.

Mark, D.M. and M.D. Gould. 1991, Interacting with Geographic Information: A Commentary: *Photogrammetric Engineering and Remote Sensing*. 57:11, pp. 1427-1430.

* * *

INTERFACE DESIGN AND KNOWLEDGE ACQUISITION FOR CARTOGRAPHIC GENERALIZATION

Harry Chang
Robert B. McMaster
Department of Geography
414 Social Sciences Building
University of Minnesota
Minneapolis, MN 55455

ABSTRACT

This paper reports on a research project that has designed a user interface for cartographic generalization that is to be used for acquiring procedural knowledge from maps. This graphical user interface consists of a series of pull-down menus, slider bars, and multiple windows to compare the results of the various generalizations. For instance, a user selecting the generalization operator, ***simplify***, is given a series of specific algorithms to select, each presented in its own window. Each window is provided with a slider bar that allows the individual to control the degree of simplification. An additional feature allows the user to apply a second operator, ***smooth***, to the data, and to overlay the original, ungeneralized, feature as well. Through the utilization of multiple windows, the user is able to compare the results of different simplification and smoothing algorithms. A last feature allows an individual to ***animate***, or generalize a feature in real time. The design provides for experimentation and enables the user to directly manipulate the tolerance values and to ascertain the quality of the generalization.

Additional operators, such as enhance, amalgamate, and aggregate, are now being added to the system. When complete, the system will allow users to generalize a database while their progress--in terms of operations and tolerance value selection--is logged. The information retrieved from these user sessions will allow the acquisition of procedural knowledge. Procedural knowledge allows control of the individual generalization operators and algorithms. The acquisition of such procedural knowledge will, ultimately, allow cartographers to build a partial knowledge base for cartographic generalization and design. The paper reports on the design of the user interface, early progress in knowledge acquisition, and the plans for development of the knowledge base.

KNOWLEDGE ACQUISITION

Several researchers, including McGraw and Harbison-Briggs (1989), have suggested several methods for acquiring knowledge, including: (1) interviewing experts, (2) learning by being told, (3) and learning by observation. **Interviewing** involves the knowledge engineer meeting with and extracting knowledge from the domain expert. In **learning by being told**, the expert "is responsible for expressing and refining the knowledge" while the knowledge engineer handles the design work (McGraw and Harbison-Briggs, 1989, p. 9). The third approach, learning by observation, is more complicated. Here the expert is allowed to interact with sample problems or case studies, or even use

previous case histories. In the field of cartography and GIS, important initial decisions will involve how, exactly, do we acquire cartographic knowledge, or, specifically, which of these techniques do we use.

Types of Knowledge

In knowledge acquisition in general, and in cartographic applications specifically, there is great difficulty in the transfer of information from the expert to the knowledge base. It has been asserted, for instance, that specific knowledge acquisition techniques should be applied to different types of knowledge. This raises the important question, "What are the general forms of human knowledge?" Three types will be discussed here: procedural, declarative, and semantic.

Procedural knowledge encompasses intrinsic human abilities including motor skills, such as walking, and mental activities, such as language. Examples of procedural knowledge from cartography include generalization operations, such as smoothing and displacement, as well as other activities including classification and interpolation in symbolization. For cartographic generalization, Armstrong has defined procedural knowledge as that "which is necessary to select appropriate operators for performing generalization tasks" (Armstrong, 1991, p. 89). As an example, certain generalization operations, such as displacement and merging, require sophisticated knowledge on feature conflict and spatial interference.

In a general sense, declarative knowledge represents the facts that we carry around. For instance, we carry thousands of bits of information, such as our street address, spouses birthday, and color of our car. In cartography, declarative knowledge would include the matrix of Bertin's visual variables, rules for dot placement, and position for type placement. Semantic knowledge, as described by McGraw and Harbison-Briggs (1989, p. 22), "represents one of the two theoretical types of long-term memory. It reflects cognitive structure, organization, and representation." Furthermore, the authors state, "Because this type of knowledge includes memories for vocabulary, concepts, facts, definitions and relationships among facts, it is of primary importance to the knowledge engineer." In cartography, semantic knowledge is often associated with the generating processes of a feature. A river, for instance, may geometrically behave much differently depending on whether it is mountainous or coastal. Thus knowledge of the geomorphological metamorphosis of a river may be necessary for both generalization and symbolization. It appears that this is similar to what Armstrong calls "structural knowledge".

It should be noted that Armstrong also identifies geometrical knowledge as feature descriptions encompassing absolute and relative locations. Geometrical knowledge, in the format of spatial primitives, is contained in the Digital Cartographic Data Standard.

It is clear from the variety of knowledge types that different techniques will be necessary for successful knowledge acquisition in cartography. In existing rule bases for symbolization and generalization, such as those reported in McMaster (1991) for the Defense Mapping Agency and Mark (1991) for the United States Geological Survey, there exists a wealth of information on geometrical and

descriptive knowledge. Transferring this factual information into a set of codified practical rules will be challenging. Buttenfield (1989, 1991) makes good progress at extracting structural or semantic knowledge using structure signatures, and knowledge from her measures will be critical in differentiating geomorphological line types and perhaps adjusting tolerance values. Other semantic information may be added at the time of encoding, such as the USGS's Enhanced Digital Line Graph data. There is a paucity of work, however, in the acquisition of procedural knowledge for cartography. As a generalized model in terms of knowledge type and extraction technique, McGraw and Harbison-Briggs (1989) present the following structure:

KNOWLEDGE	ACTIVITY	SUGGESTED TECHNIQUE
Declarative	Identifying general (conscious) heuristics	Interviews
Procedural	Identifying routine procedure/tasks	Structured interview Process Tracing Simulations
Semantic	Identifying major concepts and vocabulary	Repertory Grid Concept Sorting
Semantic	Identifying decision-making making procedures/heuristics	Task Analysis Process Tracing
Episodic	Identifying analogical problem solving heuristics	Simulations Process Tracing

[from McGraw and Harbison-Briggs (1989, p. 23)]

Those involved in building expert systems and knowledge bases in cartography and GIS will have to increasingly use innovative techniques for knowledge extraction. Simulations provide a good method for acquiring the difficult-to-extract procedural knowledge. The design of task-oriented user interfaces for procedural knowledge acquisition that allow the user to experiment and simulate specific cartographic processes appears to have great potential. One such task is cartographic generalization.

CARTOGRAPHIC GENERALIZATION

Although the process of digital cartographic generalization, a significant aspect of visualization, advanced quickly during the period 1965 - 1980, little progress has been made during the 1980s. Most of the initial progress resulted from work in the development of algorithms (such as the well known Douglas and Peucker line simplification routine, a variety of techniques for smoothing data, and algorithms for displacement, such as those by Nickerson), and attempts to analyze both the geometric and perceptual quality of those algorithms. Recent attempts at developing a more comprehensive approach to the digital generalization of map features--such as the application of simplification,

smoothing, and enhancement routines either iteratively or simultaneously--have not been, for the most part, successful (McMaster, 1989). This stems in part from our lack of procedural information--or knowledge--on generalization. Such procedural knowledge includes decisions on which techniques are applied to actually generalize map information, the sequence in which these techniques are applied, and what tolerance values, or parameters, are used. Until researchers working in the spatial sciences have access to such procedural knowledge, a comprehensive approach to cartographic generalization will not be possible.

Currently, there are no commonly available logical interfaces that enable individuals to "experiment" with map generalization and ultimately to acquire such procedural knowledge. It is, however, worth mentioning a few of the previous packages with generalization capability. Perhaps the first attempt to include generalization within a mapping package was the Harvard Laboratory's SYMAP. With SYMAP, the generalization of surfaces was possible through ELECTIVE 38: Trend Surface Analysis. The user, after constructing the required database, could select both the order of the surface and the creation of a residual map. Much later, with SAS/GRAPH, the possibility for line generalization of coordinate strings was made available. SAS/GRAPH included PROCEDURE GREDUCE which applied the Douglas (Peucker) algorithm. The algorithm was controlled with two parameters: the NX value (which represented a density level of 1-5) and the EX value (which allowed the specification of a distance tolerance). The lack of any interactive user interface required a user to iterate through the NX and EX parameters until desirable output was obtained. It was a confusing, frustrating, and time-consuming process. More recently, one can point to the interface (TRANSFORMATION WINDOW) provided by the MAP II Processor as an example of a much improved raster-based design. Here, the user selects both a MAP and a FILTERING technique from a window. After this selection, a set of specific FILTERING techniques is displayed. The user may APPLY the filtering technique, view the displayed map, and immediately select an alternative technique.

Two very important aspects of this generalization user interface include (1) an on-line (hopefully hypermedia) help system and (2) the ability to, for certain operations, perform "animated" generalization. For instance, a user could request an animated simplification of a contour line where, as the line is continually reduced, the user could stop the sequence at an acceptable level.

USER INTERFACE DESIGN

A project in the Department of Geography at the University of Minnesota involves the development of such a graphical user interface (GUI), designed specifically to gain the "procedural" knowledge involved in map generalization (Figure 1). Using a Sun SPARCstation, a user interface (designed using SUNPhigs) has been developed. The basis for the user interface is a set of multiple windows that allows the user to experiment with different simplification and smoothing algorithms. For each of the four windows created, a separate simplification algorithm is available. The user, through direct manipulation of a series of sliding bars for each window, may quickly change the scale, position of the feature, and tolerance value. It is also possible, for each of the simplification windows, to pull down a procedure window that allows:

(1) overlay of the original feature, (2) smoothing of the feature using a series of algorithms, and (3) animation. In the animated sequence the feature is simplified from the densest set of points to a caricaturized version in real time.

For instance, using window [1] a user could (1) simplify a feature using the Lang algorithm, (2) smooth the feature with a five-point moving average, and (3) overlay the original for comparative purposes. The system is now being improved with the addition of geometrical measures, such as change in angularity, and the option of stopping the animation at any time. Future development will incorporate additional generalization operators. Eventually, the user interface will be used with the generalization test data set, developed by the NCGIA, to gain procedural knowledge on generalization from trained professional cartographers, or "domain engineers". As a cartographer works with the image via the interface, a generalization "log" will be maintained. Such a log will record, for each feature, the application and sequencing of operators, along with specific tolerance values. Such knowledge will be used to determine, in a holistic manner, how maps are generalized by evaluating the relationship among operators, parameters, and features. In effect, crucial procedural information will be recorded.

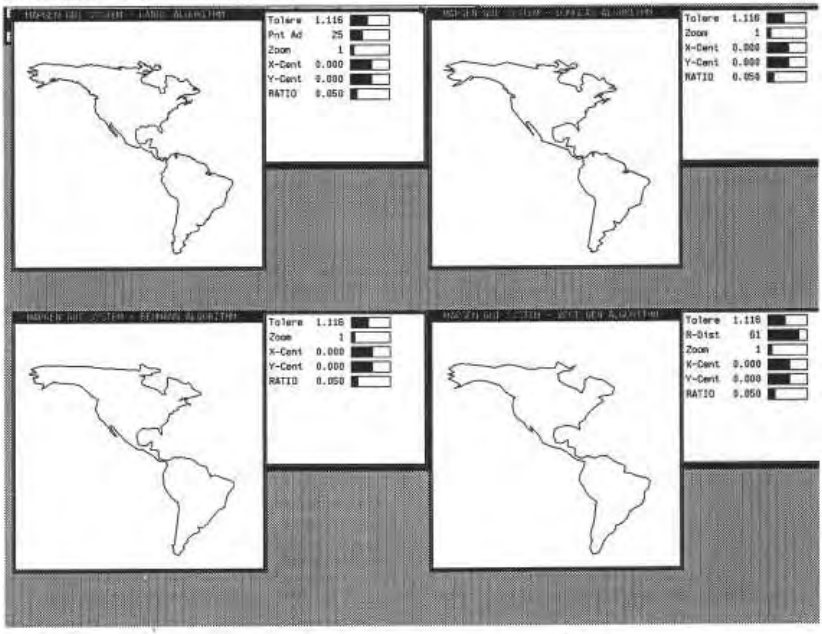


Figure 1. Generalization user interface with four simplification operators.

As GISs become more sophisticated, and the potential for quality cartographic output is finally realized, it will be necessary to pursue the application of expert systems. This will initially be in fairly narrow domains, such as typographic placement, symbolization selection, and generalization. The significant problem in the use of expert systems is in acquiring and formalizing the necessary cartographic knowledge. Capable domain and knowledge engineers must be identified, specific knowledge acquisition techniques need to be developed, and

well-thought out simulations for complex cartographic processes should be designed. In developing a research agenda, then, these are the critical initial issues that need to be addressed.

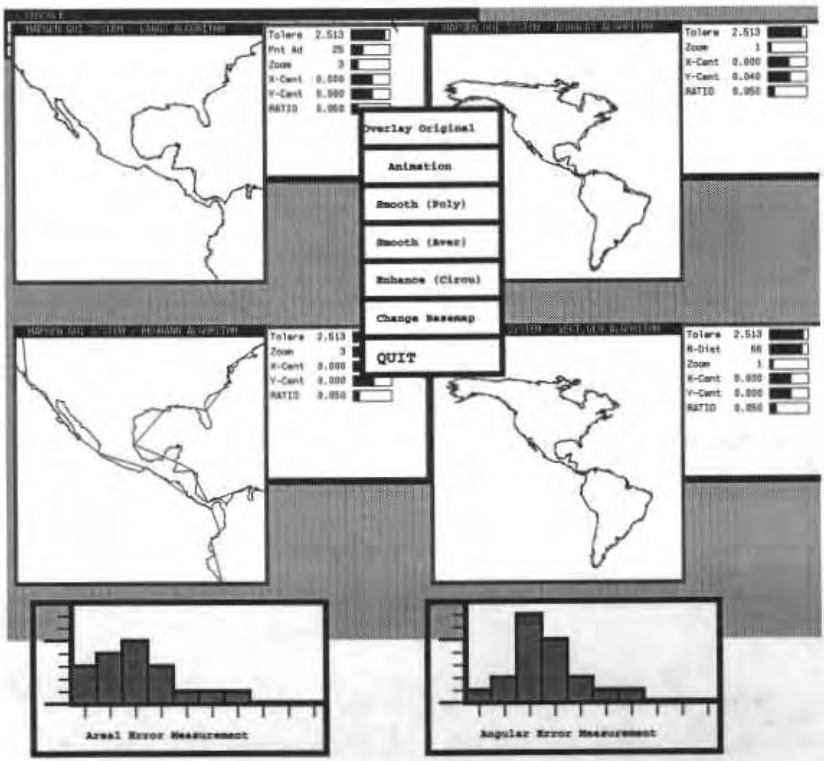


Figure 2. A variety of scales for the simplified image with simplification and original line depicted. Note the pull-down menu available for each of the original windows with selections for: (1) original overlay, (2) animation, (3) polynomial smoothing, (4) weighted-average smoothing, enhancement, and reselection of basemap.

Figure 1 depicts the interface, as copied from the Sun Workstation screen into a Postscript file. The Map Generalization System (MGS) provides the user with four windows, each with a simplification routine. It was decided, given the current level of knowledge in cartographic generalization, that most users would initiate the process through simplification. The four algorithms available include: Lang algorithm (top left), Douglas algorithm (top right), Reumann algorithm (bottom left), and Vectgen (bottom right). Descriptions of these algorithms may be found in McMaster (1987a) and are evaluated geometrically in McMaster (1987b). Originally, the interface provided six algorithms, including Jenks' and nth point, but these were later dropped for space considerations. Associated with each image window is a control window with sliding bars, including those for (1) manipulating the tolerance value, (2) zooming the image, (3) relocating the x-center and (4) y-center, and (5) a ratio value for enhancement. Some algorithms require an additional bar for control of the

simplification operator. In Figure 1, each of the four map images of the North American continent has been generalized using the same linear tolerance value: 1.116 units. Note the significantly different results of the generalizations produced by each of the algorithms. At this particular level of simplification, the Lang and Douglas routine retain more of the original detail.

Note the set of bars for the Lang algorithm. The top bar can be slid to adjust the tolerance and, in real time, the image is simplified in the image window. With Lang, the look-ahead value of points, set to 25 in this example, can also be adjusted.

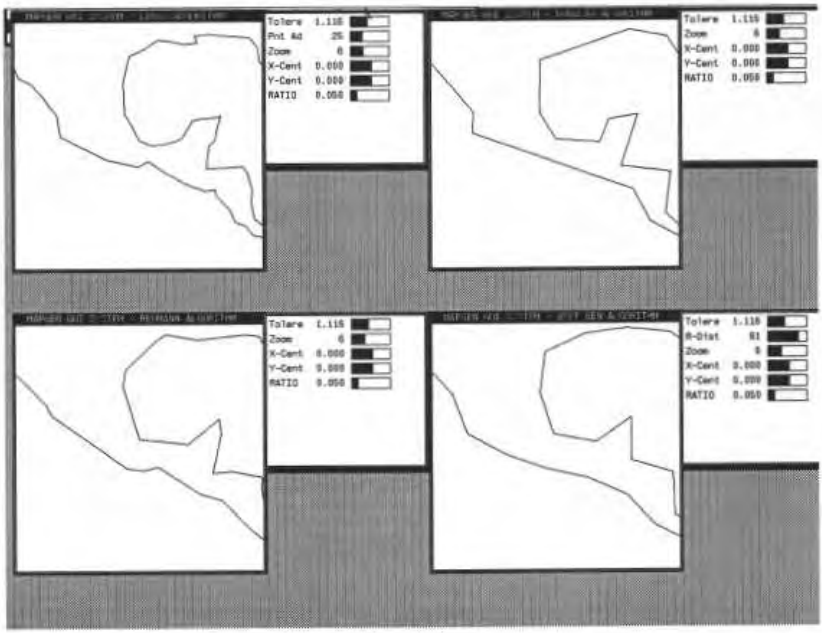


Figure 3. Use of the zoom and recenter functions for Central America.

In Figure 2, the zoom, x-center, and y-center functions have been modified to focus the image--with the same level of simplification produced by the four algorithms--on Central America. For each of the four windows, an additional menu bar may be pulled down. This menu bar allows for (1) overlay of the original line, (2) smoothing using a B-spline, (3) smoothing using a five-point weighted-moving average, (4) a simple enhancement routine, and (5) the selection of a new base map.

Figure 3 depicts a greatly enlarged (6x) view of the Central America coastline, as simplified using the same tolerance. Using the zoom and recentering functions, the user may focus in on various "problem areas". In this instance, note that Lang algorithm retains more of the original detail along the coastline at the same tolerance-level, but requires more coordinate information.

Figure 4 illustrates the overlay of the original digitized feature. This menu bar allows for the original feature to be toggled on and off for each of the windows. Figure 5 depicts a B-spline smoothing of the lines in Figure 3. Again, the B-spline can be toggled on and off. As depicted in Figure 2, the menu bar allows for an animation. Here, the user may continuously simplify a line from most to least complex in real time. An option to stop the animation at any point is now being implemented.

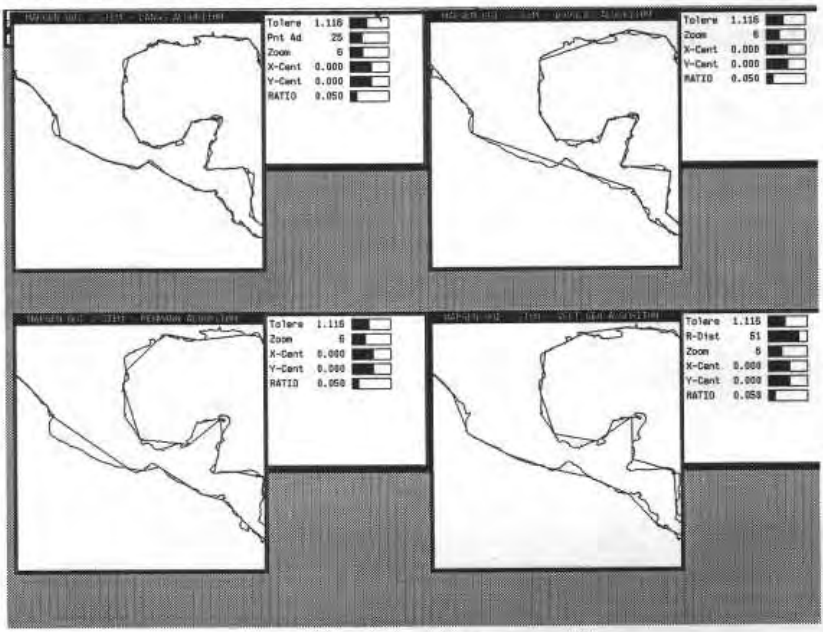


Figure 4. Overlay of the original line.

SUMMARY

This paper has reported on the development of a graphical user interface for map generalization. Such a conceptual structure was suggested in McMaster and Mark (1991). The initial work has focused on (1) the overall layout and functionality of the interface and (2) the application of simplification and smoothing operators. The user is also given the capability to overlay the original digitized feature, rescale and recenter the feature, and animate the simplification. Current work involves adding geometric measures, such as angular and areal modification, and enhancement and displacement operators.

The ultimate goal for the interface is for the acquisition of procedural knowledge on generalization. Given a map image, for instance, in what sequence would an operator apply the simplification, smoothing, and displacement operators? In order to extract such knowledge, it will be necessary to create user logs within the interface that allow for the careful recording of operator application, sequencing, and the selection of tolerance values. Such a graphical user interface, when complete, will enable cartographers to begin the process of procedural knowledge acquisition for the purposes of building knowledge bases.

Figure 5. B-spline smoothing of Figure 3, with original simplification.

Bibliography

Armstrong, Marc P. 1991. "Knowledge Classification and Organization," in Map Generalization: Making Rules for Knowledge Representation. Barbara Buttenfield and Robert B. McMaster, (eds). (United Kingdom: Longman), pp. 86-102.

Buttenfield, Barbara P. and Robert B. McMaster. 1991. Map Generalization: Making Rules for Knowledge Representation. (United Kingdom: Longman)

Buttenfield, Barbara P. 1989. "Scale-Dependence and Self-Similarity in Cartographic Lines." Cartographica, Vol. 26, No. 1, pp. 79-100.

Mark, David S. 1991. "Object Modelling and Phenomenon-Based Generalization," in Map Generalization: Making Rules for Knowledge Representation. Barbara Buttenfield and Robert B. McMaster, (eds). (United Kingdom: Longman), pp. 103-118.

McGraw, Karen L.and Karen Harbison-Briggs 1989. Knowledge Acquisition: Principles and Guidelines. (Englewood Cliffs, N.J.: Prentice Hall).

McMaster, Robert B. 1991. "Conceptual Frameworks for Geographical Knowledge," in Map Generalization: Making Rules for Knowledge Representation. Barbara Buttenfield and Robert B. McMaster (eds). (United Kingdom: Longman), pp. 21-39.

McMaster, Robert B. and David M. Mark 1991. "The Design of a Graphical User Interface for Knowledge Acquisition in Cartographic Generalization." Proceedings GIS/LIS'91. Atlanta, Georgia, pp. 311-320.

McMaster, Robert B. 1989. "The Integration of Simplification and Smoothing Algorithms in Line Generalization." Cartographica, Vol. 26, No. 1, pp. 101-121.

McMaster, Robert B. 1987a. "Automated Line Generalization." Cartographica, Vol. 24, No. 2, pp. 74-111.

McMaster, Robert B. 1987b. "The Geometric Properties of Numerical Generalization." Geographical Analysis, Vol. 19, No. 4, pp. 330-346.

Tulving, E. 1972. "Episodic and Semantic Memory." In E. Tulving and W. Donaldson, eds. Organization of Memory. (New York: Academic Press).

CONSIDERATIONS FOR THE DESIGN OF A MULTIPLE REPRESENTATION GIS

David B. Kidner[1] and Christopher B. Jones[2]

[1] Department of Computer Studies
University of Glamorgan
Pontypridd
Mid Glamorgan CF37 1DL, UK

[2] Department of Geography
University of Cambridge
Downing Place
Cambridge CB2 3EN, UK

ABSTRACT

Maintenance of a multiple representation GIS using data from a variety of sources at differing scales requires update processing that can recognise equivalences and differences between new data and stored representations. Factors that may be taken into account when establishing equivalence and difference include a) measures of similarity of the location of new and stored geometry, having regard to positional error; b) comparison of classification and names; and c) comparison of geometric shape parameters. Decisions about update of representations of equivalent real-world phenomena may depend upon the capabilities of automatic generalisation procedures that could be used to derive one scale of representation from another. Access to data at different levels of detail, for answering queries and for processing updates, is facilitated by the use of multiresolution data structures.

INTRODUCTION

A multiple representation GIS allows the same real-world phenomena to be represented in different ways, at different scales and with different levels of accuracy. Many of the problems associated with their use refer to the changes in geometric and topological structure of digital objects which occur with the changing resolution at which those objects are encoded for computer storage, analysis and depiction (NCGIA, 1989, 1993).

Many organisations concerned with spatially-referenced information are faced with the need to integrate data from a variety of sources representing phenomena at a range of scales and in some cases at a range of points in time (NCGIA, 1989, 1993). Data received from a single source, such as a national topographic mapping agency may well be accompanied by indications of changes and updates from previous versions of the data. When working with multiple source data however, problems can arise in deciding which new data items should replace existing stored data and which should

supplement existing data. When the same real world phenomena are represented at significantly different scales, decisions may need to be taken to determine whether multiple representations should be stored or whether a small scale representation could be automatically derived from a large scale representation using automatic generalisation procedures.

Ideally a GIS should include sufficient intelligence to recognise equivalences between spatial representations at different scales and different times and either automatically make decisions on appropriate update strategies, according to pre-specified rules, or else assist the user in making decisions by highlighting equivalences and differences between representations.

In a large database, efficiency in storage and access to multiscale and multiple representation data can be provided by multiresolution data structures providing progressive access to increasing levels of detail. If data for different themes and scales are to be combined in a flexible manner, it is also desirable to provide facilities for automated generalisation whereby appropriate levels of detail are selected and representations are modified to suit particular purposes and to avoid visual conflict between cartographic symbology.

An overview of a possible database architecture to support such a multiscale, multiple representation GIS was provided by Jones (1991), in which it was proposed that a deductive database architecture might be employed in combination with multiresolution data structures and specialised processors for performing update and scale-variable retrieval. In this paper we focus in more detail on some of the issues that arise in attempting automatic update of multiscale data that may be stored in multiple representations and with multiresolution data structures. Resulting strategies for update are being used to develop an experimental update processor for multiscale databases. This is being implemented using object-oriented programming with rule-processing.

STORAGE STRATEGIES FOR MULTIPLE REPRESENTATIONS

In a multiple representation database a many to many relationship may be established between real-world object descriptions, in the form of classifications and named and uniquely identified phenomena, and geometric-object descriptions consisting of sets of spatial primitives. Since real-world phenomena may be described at varying levels of generalisation, it is possible to envisage considerable complexity with multiple levels of overlapping, hierarchical, real-world object representations. Thus, for example, different types of overlapping administrative areas may refer to common lower level regions or topographic features. The different levels of each real world object hierarchy could refer to geometric objects that represented them spatially. Thus a high level real-world feature could have a relatively simple representation, such as a polygon, as well as the more detailed representations of its constituent parts. Individual items of geometry could be referred to from the various real-world objects that they

were used to represent. Conversely, each geometric object can refer to the real-world objects that it is used to represent.

When extensive geometric objects are represented with relatively high accuracy, they may be subject to generalisation operators on retrieval. Efficiency in creating generalised representations can be gained by organising data by means of multiresolution data structures (e.g. van Oosterom, 1990, 1991; Becker et al, 1992; Jones and Abraham, 1986, 1987; Ware and Jones, 1992). Most of these data structures implemented to date make use of line or surface simplification procedures to categorise constituent vertices with different levels of scale significance.

In the Multi-Scale Line Tree (MSLT) of Jones and Abraham (1986, 1987), vertices of linear geometry are classified within error bands as defined by the Douglas algorithm (Douglas and Peucker, 1973). The resulting levels are spatially indexed using a quadtree scheme. Becker et al (1992) employ a related approach in which each level is spatially accessed by an R-tree type spatial indexing mechanism that accesses short sections of linear geometry. In the Reactive-tree, van Oosterom (1991) organises the vertices of entire line segments in a hierarchy based on classification of the vertices using the Douglas algorithm. Each complete line is given a priority and is spatially indexed with an R-tree type data structure. In the Multiscale Topographic Surface Database (MTSD), Ware and Jones (1992) integrate point, line and polygon objects within a hierarchically-structured terrain model based on constrained Delaunay triangulation. The vertices of objects embedded in the surface are classified by a combination of the Douglas algorithm, providing lateral error control, and the surface simplification algorithm used by De Floriani (1989) for vertical error control. In a recent development of the MTSD, flexibility in access is combined with considerable data storage savings by using the Implicit TIN approach to triangulated surface storage (Kidner and Jones, 1991), in which only vertex coordinates are stored explicitly, while the topology of the constrained triangulation is determined at the time of retrieval (Jones, Kidner and Ware). Using this approach, in which no explicit triangulation data are stored, the surface can be constrained with only those features that are relevant to the particular user query.

Multiresolution data structures of the sort referred to have been criticised for their dependence upon algorithms such as that of Douglas and Peucker (1973), which cannot be guaranteed to provide cartographically sound generalisations (Visvalingam, 1990). In practice, by building the data structure by means of algorithms that classify constituent vertices progressively according to their contribution to increasing locational accuracy, the data structure is able to provide efficient access to a representation that at least satisfies quantitative criteria relating to resolution. Generalisation may be regarded as a separate process that operates upon the retrieved geometry. Clearly the data structures enable as much of the stored geometry to be retrieved as is required by the generalising processor, which may then transform the retrieved data. The distinction is one that is made by Brassel and Weibel (1988) for example, who emphasise the importance of separating a Digital Land Model (DLM), based on survey

data, from a Digital Cartographic Model, that represents a transformation of DLM data to a form suitable for effective visualisation. (This is not to argue however that there may not be algorithms that are preferable to that of Douglas and Peucker in the ranking order that is placed upon the vertices of a line.)

METADATA

Effective storage and maintenance of multiple representations will depend upon recording metadata relating both to real-world and geometric descriptions. If data originate from several sources, a record must be maintained of the source with regard to the types of classification and nomenclature schemes, along with dictionaries of codes used to describe the real-world attributes. 'Metadata relating to geometry include the positional accuracy of the survey method, the precision of recorded coordinates, the date of original survey, the coordinate system used, the geodetic datum, any transformations known to have been applied to the data and, specifically for data acquired by digitising existing maps, the map series, the scale of the map, the map projection, the positional error in digitising and the date of production of the map.

Appropriate interpretation of metadata relating to classification and nomenclature could depend upon the storage of rules that encode the hierarchical nature of individual classification schemes and the relationships between particular categories or types in different schemes. Rules of this sort are required to assist in determining the equivalences and differences between new data and stored data. Deductive databases can provide the facility to encode and execute such rules (Jones, 1993).

UPDATE SITUATIONS IN A MULTIPLE REPRESENTATION DATABASE

Maintenance of multiple versions of scale-variable data introduces considerable complexity to the data update procedures. Operations that may take place on update may be reduced in principle to those of adding a new representation and of deleting an existing representation. However, the word representation here must be interpreted as covering the range between detailed geometric coverages of a region of space and individual geometric primitives representing either localised phenomena or generalised descriptions of extensive phenomena. Furthermore, addition of data may involve attempts to unify the new data with existing data both geometrically, by edge matching, and semantically by recording situations in which the new geometry represents the same phenomena as does an existing representation.

Decisions about whether to add or delete scale-variable representations will depend on several factors. In a multiscale database, one of the most important of these factors is that of whether one representation can easily be derived from another. With the current status of generalisation software there is still very limited capacity to perform generalisation entirely automatically and thus there is often justification for storing versions of data

at various levels of generalisation. It may be remarked though, that when integrating different datasets at retrieval time it would probably, never in practice be possible to have access to all possible pre-generalised representations, since the nature of generalisation may depend upon graphic interaction between specific items of data that may vary from one query to the next. At present there is certainly some limited capacity for entirely automatic generalisation, in particular for linear feature simplification and for scale-dependent selection of data, and this capacity could be exploited in a multiple representation GIS. To do so however could require that quantitative limits were placed on the ranges of scales over which generalisation could take place. This would allow update decisions to be taken to justify multiple representation.

Another major factor affecting the decision to store multiple representations is a knowledge of whether new data are equivalent to existing stored data. If data can be shown to be equivalent in that they represent the same real-world phenomena, then storage of multiple representations can be addressed from the previous point of view of whether or not existing data can be derived from other data automatically. If new data items can be proven to be separate in location from existing data then it may be assumed that they should be stored. As indicated earlier, the problem of establishing equivalence is most likely to arise when working with data from multiple sources, that may be at different levels of generalisation. Thus some datasets may record the presence of features that were omitted in another dataset, due to some process of feature selection. Equally, time-variant datasets may record differing features, such as administrative boundaries, roads and buildings, due to modifications or new developments.

The process of recognition of equivalence and difference is complicated by the fact that geometric location is always accompanied by uncertainty, the degree of which will vary between data recorded at different scales. Before considering methods of determining equivalence and difference we summarise below some of the factors that must be taken into account.

Differences in accuracy and generalisation between new data and stored data:

- new data may be less generalised than a stored representation
- new data may be equivalently generalised to a stored representation
- new data may be more generalised than a stored representation

The ability to derive one representation from another:

- new data may be derivable from a stored representation
- new data may be used to derive a stored representation
- new data may not be related to a stored representation

The locational relationships between new data and stored representations, all of which are established to some level of certainty:

- an entire new dataset may be separate from stored representations
- an entire new dataset may overlap a stored representation
- an entire new dataset may be edge-adjacent to a stored representation

- an entire new dataset may be partially edge-adjacent to a stored representation

- a new geometric primitive may be equivalent in location to an existing stored primitive
- a new geometric primitive may be separate from stored primitives
- a new geometric primitive may be partially equivalent to stored primitives
- a new geometric primitive may be edge-adjacent to stored primitives

The classification relationships between new and stored representations:

- the classification of new real-world objects may be the same as those of stored objects
- the classification of new real-world objects may be different from those of stored objects
- the classification of new real-world objects may be similar within an established level of certainty to stored objects.

TECHNIQUES FOR ESTABLISHING EQUIVALENCE AND DIFFERENCE

A key issue that emerges from consideration of the above factors is that of establishing equivalence of geometry that must be regarded as having inherent positional error. Since all stored geometry must be regarded as representing some real-world phenomena, methods for establishing equivalence can attempt to exploit both semantic and geometric data. Certain pieces of geometry may represent two or more real-world phenomena. Thus a single line segment could represent both the centre of a river and an administrative boundary. The same line would be less likely to represent a railway. Evidence for equivalences can be built up on the basis of multiple criteria that include the classification, the location and the shape characteristics.

Locational equivalence needs to encompass the situations of complete overlap, partial overlap (in which case the equivalent part must be identified) and adjacency, in which it must be determined whether two pieces of geometry are continuous, i.e. two parts of the same real-world phenomenon.

When comparing the location of geometric objects, it is essential that they are regarded as fuzzy in the sense that all coordinate data have associated errors. Error and uncertainty have always been a feature of cartographic information. However, the problem of handling error is compounded further within a multiple representation GIS. Source accuracy indices, metadata and functions are required to propagate these values to document the quality of operations (Lanter and Veregin, 1990). The uncertainty of digital map features may be characterised in terms of regions of locational probability around their edges, which are adaptable depending upon the level of certainty with which one wishes to access fuzzy objects (Dutton, 1992, Maffini et al, 1989). Brunsdon et al (1990) present a review of methods for

handling error propagation, including the traditional 'epsilon' approach and a technique based on Monte Carlo simulation.

A variety of techniques are available to assist in determining whether two fuzzy spatial objects appear to be equivalent in location. The least computationally demanding method of determining separateness is that of comparison of the extents of the objects as indicated by minimum bounding rectangles. To give the method some reliability, the extents must be expanded to take account of the maximum error associated with the coordinates. If that test failed to establish separateness, in the case of a line, an extent oriented parallel to the 'anchor line' connecting start and end points could be used. If this resulted in overlap, then buffer zones could be created around each object and their intersections determined. The percentage overlaps relative to each object would constitute a measure of equivalence, though interpretation of this measure would depend upon several factors that include the relative dimension of the two objects and the magnitude of errors assumed for each object. Determination of the nature of partial equivalence of lines would require comparison of buffers associated with constituent vectors.

Analysis of levels of certainty of locational overlap using vector defined buffers could be expected to result in considerable computational overheads. These overheads could be reduced somewhat by working with simplified versions of the objects to be compared. Thus for example a line simplification algorithm might be used to reduce both objects to a similar level of detail. If stored data were represented by a multiresolution data structure, the simplified version of the line could be readily accessed, along with the corresponding error band. The method is thus comparable to that of operations upon strip trees (Ballard, 1981), though the assumption here is that buffer zones would take account of positional error, rather than just the error tolerance used in a simplification algorithm.

Alternative methods of location-based comparison could employ raster representations of the geometric objects. If rasterisation was performed to a relatively high resolution, then each pixel could be associated with a certainty measure, based on a probability distribution function centred on the stored geometry. When the raster representations were overlaid, these measures could then be combined to establish levels of certainty of overlap.

Locational comparison methods can provide valuable evidence of the equivalence of geometry. However, when the certainty level of comparisons is not very high, for example, due to a systematic locational shift, other methods, that characterise the geometry of objects, may also be considered to help accumulate evidence for similarity or difference. A major difference in geometric signature might be used to establish difference, without performing more detailed locational comparisons. Below are listed some examples of geometric parameters. Some of these are based on the line signature parameters proposed by Buttenfield (1991), and McMaster's (1986) measures for evaluating linear simplification.

1. Minimum bounding rectangle dimensions.
2. Line length.
3. Anchor line length, measuring the Euclidian distance between the first and last coordinates of a line.
4. Bandwidth, based on the maximum perpendicular deviations of coordinates from the anchor line. Bandwidth may also be standardised by division by the anchor line length.
5. Segmentation, which is the displacement along the anchor line of the point of maximum perpendicular distance.
6. Error variance, which is the square root of the sum of the squared distances of line vertices from their corresponding anchor line, divided by the number points.
7. Concurrence, which is related to the number of times that a line crosses its anchor line.
8. Sinuosity ratio, which is the ratio between the line length and the anchor line length.
9. Average angular change, which may be based on the angles between successive vectors defining a line.
10. Curvilinearity, which represents the tendency for successive line vectors to change direction positively or negatively.
11. Average separation of successive vertices in a line.
12. Number of vertices in a line.
13. Polygonal area.
14. Perimeter/Area ratio.

In attempting to use parameters of the sort listed above, it must be appreciated that some of them are scale-dependent. Thus when comparing lines from different scale representations, less, or no, emphasis should be placed on parameters that depended on the number of vertices, on measurements of line length or on frequency of directional change. It is however possible to compensate for scale differences. One approach is to apply factors that might be a function of line sinuosity or fractal dimension. Another is to simplify the more detailed representation in order to render the levels of generalisation comparable. In the case of line segments, this could be done with a line simplification algorithm or by retrieval from a multiresolution data structure, as indicated above in the context of locational comparison.

A RULE-BASED OBJECT-ORIENTED SYSTEM FOR MULTISCALE DATABASE UPDATE

Work is currently underway on the development of a rule-based system to control update in a multiscale, multiple representation database. It is implemented with an object-oriented programming system that includes rule processing. Programming objects that are employed include those of

update dataset objects, containing both geometry and classification data; real-world objects equivalent to entries in a permanent database; and geometric objects also corresponding to permanently stored data. All of these object classes include slots to store relevant metadata derived from a permanent database. The constituents of dataset objects are matched against real world and geometric objects in an attempt to establish equivalences and differences between them and to create new real-world and geometric objects that may subsequently be loaded to the permanent database.

Methods are being implemented to calculate geometric shape parameters that will be stored, if calculated, within relevant slots of the geometric objects and the dataset objects. Locational comparison methods are also being implemented, based initially on vector-based techniques. Experiments are being carried out using rule processing to control execution of the location and shape descriptor methods and to compare classifications. The rules use results of the comparisons to accumulate measures of equivalence between components of new datasets and database representations and make decisions about database update operations of addition and deletion.

CONCLUSIONS

This research has been aimed at deriving new methods for handling multiple representations within a GIS. These aims are in accordance with the views of others tackling the same problem (NCGIA, 1989, 1993), namely, that two main areas in which research should be focused are a) database issues, such as the need to organise multiple topological and metrical versions for efficient access; and the implementation of linkages between multiple representations; and b) generalisation issues, such as formalising digital feature description models, flexible definitions of resolution for data sets, and rules for map generalisation. This paper addresses these issues with respect to maintaining a multiple representation GIS for update processing in particular.

The process of updating a GIS database using multisource data is complicated by problems of recognising the equivalence and difference between new data and that already stored. Determination of equivalence is subject to error due to locational inaccuracy at varying scales and to the possibility of differing classification schemes. A rule-based approach is being adopted in an experimental system to evaluate strategies for multiscale database update. The rules make use of methods that implement geometric procedures for comparing location and for comparing a variety of shape descriptors. The system is being developed in the context of a database capable of providing multiresolution access to stored geometry.

ACKNOWLEDGEMENTS

This research is supported by grant GR/F96288 from the Science and Engineering Research Council and by the Ordnance Survey, Southampton.

REFERENCES

Ballard, D.H. (1981) "Strip Trees: A Hierarchical Representation For Curves", Comm. of the ACM, Vol. 24(5), May, pp. 310-321.

Becker, B., H. Six & P. Widmayer (1991) "Spatial Priority Search: An Access Technique for Scaleless Maps", Proc. of the 1991 ACM SIGMOD, Denver, Colorado, May 29-31, (Eds: J. Clifford & R. King), ACM SIGMOD Record, Vol. 20(2), June, pp. 128-137.

Brassel, K.E. & R. Weibel, (1988), "A Review and Conceptual Framework of Automated Map Generalization", Int. Journal of Geographical Information Systems, Vol. 2(3), pp. 229-244.

Brunsdon, C., S. Carver, M. Charlton & S. Openshaw (1990) "A Review of Methods for Handling Error Propagation in GIS", Proc of the 1st European Conference on GIS (EGIS'90), Amsterdam, Netherlands, pp. 106-113.

Buttenfield, B.P. (1991) "A Rule for Describing Line Feature Geometry", in Map Generalization: Making Rules for Knowledge Representation, (Eds: B.P. Buttenfield & R.B. McMaster), Longman, London, U.K., pp. 150-171.

De Floriani, L. (1989) "A Pyramidal Data Structure for Triangle-Based Surface Description", IEEE Computer Graphics & Applications, March, pp. 67-78.

Douglas, D.H. & T.K. Peucker (1973) "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature", The Canadian Cartographer, Vol. 10(2), December, pp. 112-122.

Dutton, G. (1992) "Handling Positional Uncertainty in Spatial Databases", Proc. of the 5th Int. Symp. on Spatial Data Handling, Charleston, USA, pp. 460-469.

Jones, C.B. & I.M. Abraham (1986) "Design Considerations for a Scale-Independent Cartographic Database", Proc. of the 2nd Int. Symp. on Spatial Data Handling, Seattle, Washington, July 5-10, pp.384-398.

Jones, C.B. & I.M. Abraham (1987) "Line Generalisation in a Global Cartographic Database", Cartographica, Vol. 24(3), pp.32-45.

Jones, C.B. (1991) "Database Architecture for Multi-Scale GIS", Proc. of the 10th Int. Symp. on Computer Assisted Cartography (Auto-Carto 10), ACSM/ASPRS, Baltimore, Maryland, March 25-28, pp.1-14.

Jones, C.B., "Deductive Databases and the Representation of Hierarchies in Geographical Information", submitted for publication.

Jones, C.B., D.B. Kidner & J.M. Ware, "The Implicit TIN and multiscale spatial databases", submitted for publication.

Kidner, D.B. & C.B. Jones (1991) "Implicit Triangulations for Large Terrain Databases", Proc. of the 2nd European Conference on GIS (EGIS'91), Brussels, Belgium, pp. 537-546.

Lanter, D.P. & H. Veregin (1990) "A Lineage Meta-Database Program for Propagating Error in Geographic Information Systems", Proc. of GIS /LIS'90, pp. 144-153.

Maffini, G., M. Arno & W. Bitterlich (1989) "Observations and Comments on the Generation and Treatment of Error in Digital GIS Data", in Accuracy of Spatial Databases, (Eds: M. Goodchild & S. Gopal), Taylor & Francis, London, pp. 55-67.

McMaster, R.B. (1986) "A Statistical Analysis of Mathematical Measures for Linear Simplification", The American Cartographer, Vol. 13(2), pp. 103-116.

NCGIA (1989) "Multiple Representations", Scientific Report for the Specialist Meeting of the NCGIA Research Initiative 3, (Eds: B.P. Buttenfield & J.S. DeLotto), Dept. of Geography, SUNY at Buffalo, Feb. 18-21, Report 89-3, 87 pages.

NCGIA (1993) "Multiple Representations", Closing Report for NCGIA Research Initiative 3, (Ed: B.P. Buttenfield), Dept. of Geography, SUNY at Buffalo, April, 27 pages.

van Oosterom, Peter (1990) "Reactive Data Structures for Geographic Information Systems", Ph.D. thesis, Dept. of Computer Science, Leiden University, The Netherlands, 197 pages.

van Oosterom, Peter (1991) "The Reactive-Tree: A Storage Structure for a Seamless, Scaleless Geographic Database", Proc. of the 10th Int. Symp. on Computer-Assisted Cartography (Auto-Carto 10), ACSM/ASPRS, Vol.6, Baltimore, Maryland, March 25-28, pp.393-407.

Visvalingam, M. & J.D Whyatt (1990), "The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation through Visualisation", Computer Graphics Forum, Vol. 10(3), pp. 225-235.

Ware, J.M. & C.B. Jones (1992), "A Multiresolution Topographic Surface Database", Int. Journal of Geographical Information Systems, Vol. 6(6), pp. 479-496.

FROM COMPUTER CARTOGRAPHY TO SPATIAL VISUALIZATION: A NEW CARTOGRAM ALGORITHM

BIBLIOGRAPHIC SKETCH

Daniel Dorling currently holds a British Academy Fellowship at the University of Newcastle upon Tyne. His research interests include studying the geography of society, politics and housing; visualization, cartography and the analysis of censuses. After the completion of his PhD (entitled 'the visualization of spatial social structure') in 1991, he was awarded a Rowntree Foundation Fellowship. He graduated from Newcastle University in 1989 with a first class degree in Geography, Mathematics and Statistics.

Daniel Dorling
Department of Geography
University of Newcastle upon Tyne
England, NE1 7RU

ABSTRACT

Computer cartography is developing into spatial visualization, in which researchers can choose what they wish to see and how they wish to view it. Many spatial distributions require new methods of visualization for their effective exploration. Examples are given from the writer's work for the preparation of a new social atlas of Britain, which not only uses new statistics but employs radically different ways of envisioning information to show those statistics in a new light - using area cartograms depicting the characteristics of over ten thousand neighbourhoods simultaneously.

INTRODUCTION

> *Suppose that one could stretch a geographical map so that areas containing many people would appear large, and areas containing few people would appear small . .*
> *Tobler, 1973, p.215*

Suppose, now, that one could stretch a geographical map showing the characteristics of thousands of neighbourhoods such that each neighbourhood became visible as a distinct entity. The new map would be an area cartogram (Raisz 1934). On a traditional choropleth map of a country the shading of the largest cities can be identified only with difficulty. On an area cartogram every suburb and village becomes visible in a single image, illuminating the detailed geographical relationships nationally. This short paper presents and illustrates a new algorithm to produce area cartograms that are suitable for such visualization; and argues why cartograms should be used in the changing cartography of social geography.

Equal population cartograms are one solution to the visualization problems of social geography. The gross misrepresentation of many groups of people on conventional topographic maps has long been seen as a key problem for thematic cartography (Williams 1976). From epidemiology to political science, conventional maps are next to useless because they hide the residents of cities while massively overemphasising the characteristics of those living in the countryside (Selvin et al 1988). In mapping social geography we should represent the population equitably.

Visualization means making visible what can not easily be imagined or seen. The spatial structure of the social geography of a nation is an ideal subject for visualization as we wish to grasp simultaneously the detail and the whole picture in full. A population cartogram is the appropriate base for seeing how social characteristics are distributed spatially across people rather than land. Although the problems of creating more appropriate projections have emerged in many other areas of visualization (see Tufte 1990, Tukey 1965).

THE ALGORITHM

Cartograms have a longer history than the conventional topographic maps of today, but only in the last two decades have machines been harnessed to produce them (see for instance Tobler 1973, Dougenik et al 1985). Most cartograms used today are still drawn by hand because the cartographic quality of automated productions was too poor or could not show enough spatial detail. A key problem for visualization is that the maintenance of spatial contiguity could result in cartograms where most places were represented by strips of area too thin to be seen. This paper deals with non-continuous area cartograms (following Olson 1976) where each place is represented by a circle. The area of each circle is in proportion to the place's population and each circle borders as many of the place's correct geographical neighbours as possible (see Härö 1968).

The Pascal implementation of the algorithm is included as an appendix so that detailed cartograms can be produced for countries other than Britain. The algorithm begins by positioning a circle at the centroid of each place on a land map and then applies an iterative procedure to evolve the desired characteristics. All circles repel those with which they overlap while attracting those with whom they share a common border. Many more details are given in Dorling (1991). ***Figure 1*** shows the evolution of a cartogram of the 64 counties and regions of Britain using this algorithm - the areas, as circles, appear to spring into place. ***Figures 2 to 6*** illustrate various graphical uses to which the cartogram can be put, ranging from change and flow mapping, to depicting voting swings by arrows or the social characteristics of places with a crowd of Chernoff faces (the cartogram is also useful when animated, see Dorling 1992).

The true value of this new algorithm is not in producing cartograms of a few hundred areas, as manual solutions and older computer programs can already achieve this. A projection has never been drawn before, however, which can clearly make visible the social structure of thousands of neighbourhoods on a few square inches of paper. ***Figures 7 and 8*** use an equal land area map to show administrative boundaries while ***Figures 9 and 10*** show the same boundaries on a population cartogram. Each of the ten thousand local neighbourhoods (called wards) are visible on the cartogram and there is enough space to name the cities which can only be shown by dots on a conventional map of British counties.

Figures 11 and 12 show the ward cartogram being used to illustrate the spatial distribution of ethnic minorities in Britain. On the ward map it appears that almost everyone is white, with the most significant feature being two *ghettos* in the mountains of Scotland. This map is completely misleading, as are all maps of social geography based on an equal land area projection. Most people in Britain live in neighbourhoods which contain residents belonging to ethnic minorities. Their most significant concentrations are in Birmingham, Leicester, Manchester, Leeds and three areas of London, where "minorities" comprise more than a quarter of some inner city populations. Conventional maps are biased in terms of whose neighbourhoods they conceal.

The new algorithm has been used to create cartograms of over one hundred thousand areal units. To show social characteristics effectively upon these requires more space than is available here and also the use of colour (see Dorling 1992). ***Figures 13 and 14*** have used such a cartogram as a base to illustrate the spatial distribution of people in Britain following the method used by Tobler (1973) for the United States. Once a resolution such as this has been achieved, the cartogram can be viewed as a continuous transform and used for the mapping of incidences of disease or, for instance, the smooth reprojection of road and rail maps. At the limit — were each areal unit to comprise of the space occupied by a single person — the continuous and non-continuous projections would become one and the same.

Population area cartograms illuminate the most unlikely of subjects. Huge flow matrices can be envisioned with ease using simple graphics programming. ***Figure 15*** shows over a million of the most significant commuting flows between wards in England and Wales. The vast majority of flows are hidden within the cities. ***Figure 16*** reveals these through reprojecting the lines onto the ward cartogram. On the cartogram movement is everywhere and so the map darkens with the concentration of flows. Just as all that other commuters can see is commuters, so too that is all we can see on the cartogram. Equal population projections are not always ideal.

The algorithm used to create these illustrations is included as a two page appendix. The author hopes that it will be used by other researchers to reproject the maps of other countries - using the United States's counties or the communes of France for example. The program requires the contiguity matrix, centroids and populations of the areas to be reprojected. It produces a transformed list of centroids and a radius for the circle needed to represent each place (its area being in proportion to that place's population). The cartograms shown here were created and drawn on a microcomputer costing less than $800.

CONCLUSION

The creation and use of high resolution population cartograms moves computer cartography towards spatial visualization. The age old constraints that come from conventional projections are broken as we move beyond the paper map to choose what and how we wish to view the spatial structure of society (Goodchild 1988). Conventional projections are not only uninformative, they are unjust — exaggerating the prevalence of a few people's lifestyles at the expense of the representation of those who live inside our cities, and hence presenting a bias view of society as a whole. If we wish to see clearly the detailed spread of disease, the wishes of the electorate, the existence of poverty or the concentration of wealth, then we must first develop a projection upon which such things are visible. The algorithm presented here creates that projection.

REFERENCES

Dorling, D. (1991) The visualization of spatial social structure, unpublished PhD thesis, Department of Geography, University of Newcastle upon Tyne.

Dorling, D. (1992) Stretching space and splicing time: from cartographic animation to interactive visualization, Cartography and Geographic Information Systems, Vol.19, No.4, pp.215-227, 267-270.

Dougenik, J.A., Chrisman NR. & Niemeyer, D.R. (1985) An algorithm to construct continuous area cartograms, Professional Geographer, Vol.37, No.1, pp.75-81.

Goodchild, M.F. (1988) Stepping over the line: technological constraints and the new cartography, The American Cartographer, Vol.15, No.3, pp.311-319.

Härö, A.S. (1968) Area cartograms of the SMSA population of the United States, Annals of the Association of American Geographers, Vol.58, pp.452-460.

Olson, J. (1976) Noncontiguous area cartograms, The Professional Geographer, Vol.28, pp.371-380.

Selvin, S., Merrill, D.W., Schulman, J., Sacks, S., Bedell, L. & Wong, L. (1988) Transformations of maps to investigate clusters of disease, Social Sciences in Medicine, Vol.26, No.2, pp.215-221.

Raisz, E. (1934) The rectangular statistical cartogram, The Geographical review, Vol.24, pp.292-296.

Tobler, W.R. (1973) A continuous transformation useful for districting, Annals of the New York Academy of Sciences, Vol.219, pp.215-220.

Tufte, E.R. (1990) Envisioning information, Graphics Press, Cheshire, Connecticut.

Tukey, J.W. (1965) The future process of data analysis, Proceedings of the tenth conference on the design of experiments in army research development and testing, report 65-3, Durham NC., US army research office, pp.691-725.

Williams, R.L. (1976) The misuse of area in mapping census-type numbers, Historical Methods Newsletter, Vol.9, No4, pp.213-216.

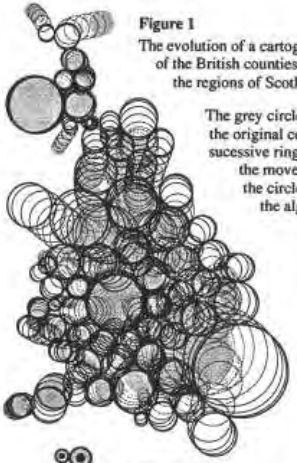


Figure 1
The evolution of a cartogram of the British counties and the regions of Scotland.

The grey circles lie on the original centroids, sucessive rings depict the movement of the circles under the algorithm.

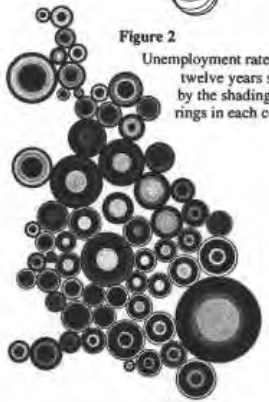


Figure 2
Unemployment rates over twelve years shown by the shading of 12 rings in each county.

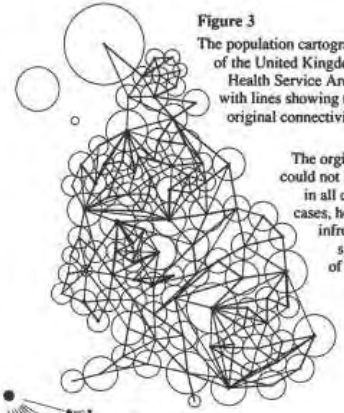


Figure 3
The population cartogram of the United Kingdom Health Service Areas with lines showing the original connectivity.

The orginal contiguity could not be maintained in all cases; difficult cases, however, occur infrequency given sufficient areas of roughly equal populations.

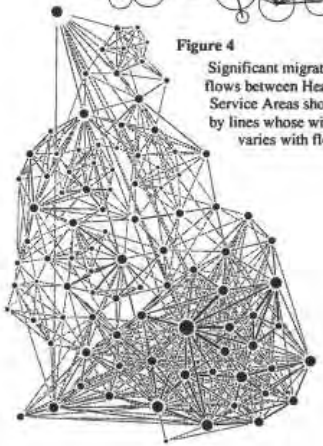


Figure 4
Significant migration flows between Health Service Areas shown by lines whose width varies with flow.

Figure 5
The swings of votes between three parties over two elections in 650 constituencies shown by arrows giving the direction and magnitude of swings on a population cartogram.

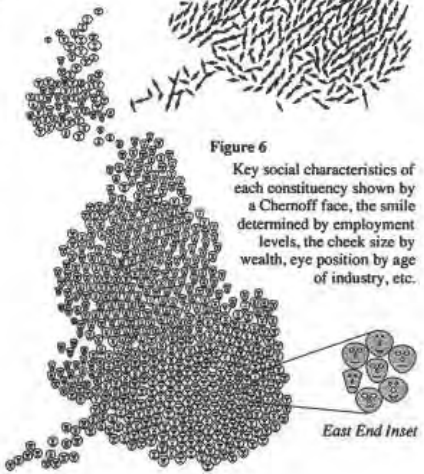


Figure 6
Key social characteristics of each constituency shown by a Chernoff face, the smile determined by employment levels, the cheek size by wealth, eye position by age of industry, etc.

East End Inset

Figure 7

Ward boundaries on an equal area map projection

The vast majority of wards are simply not visible on a conventional map no matter how large it is drawn or how many insets are included. Those areas which can be seen usually contain the least people while those which are not visible conceal the largest populations.

Figure 8

County boundaries

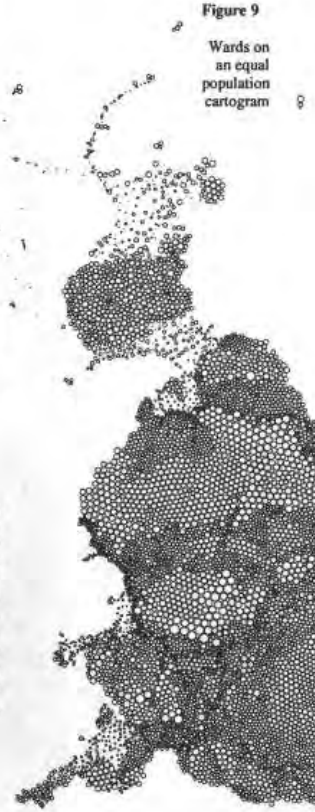


Figure 9

Wards on an equal population cartogram

On the equal popualtion cartogram each ward is shown by a circle the area of which is in proportion to its population. The reprojected county boundaries that result are shown in Figure 10. City wards tend to contain more people and hence are larger. The circles do not coalesce in the least populated areas of Scotland, Wales and the South West.

Figure 10

County boundaries around wards on the cartogram

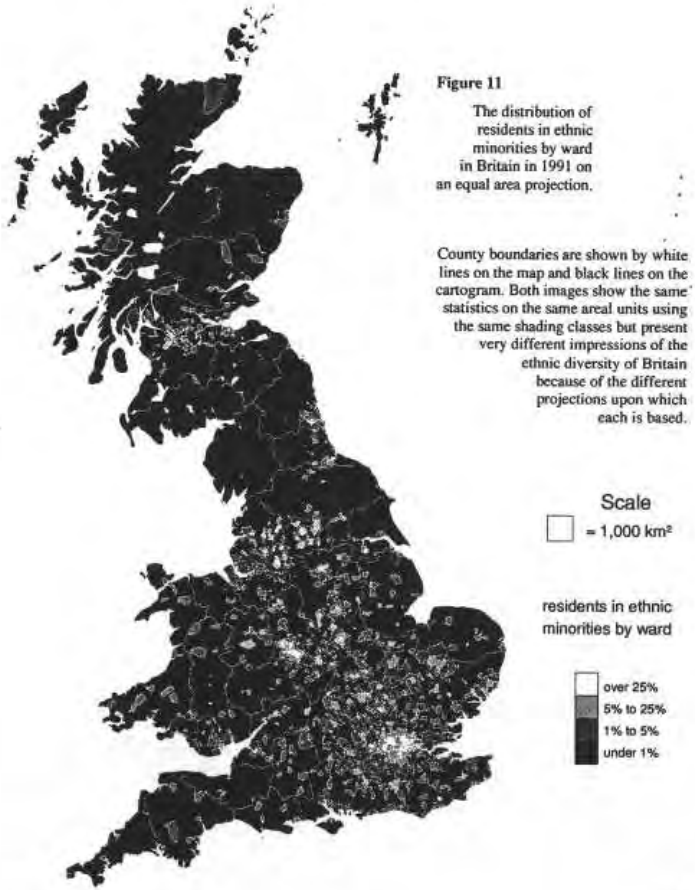


Figure 11

The distribution of residents in ethnic minorities by ward in Britain in 1991 on an equal area projection.

County boundaries are shown by white lines on the map and black lines on the cartogram. Both images show the same statistics on the same areal units using the same shading classes but present very different impressions of the ethnic diversity of Britain because of the different projections upon which each is based.

Figure 10

The distribution of residents in ethnic minorities by ward in Britain in 1991 on an equal population cartogram.

The cartogram allows the sizes of each of Britain's major ethnic minority areas to be compared and their shapes contrasted. Sharp divides are made evident while the areas in which nobody lives are not allowed to obscure the image.

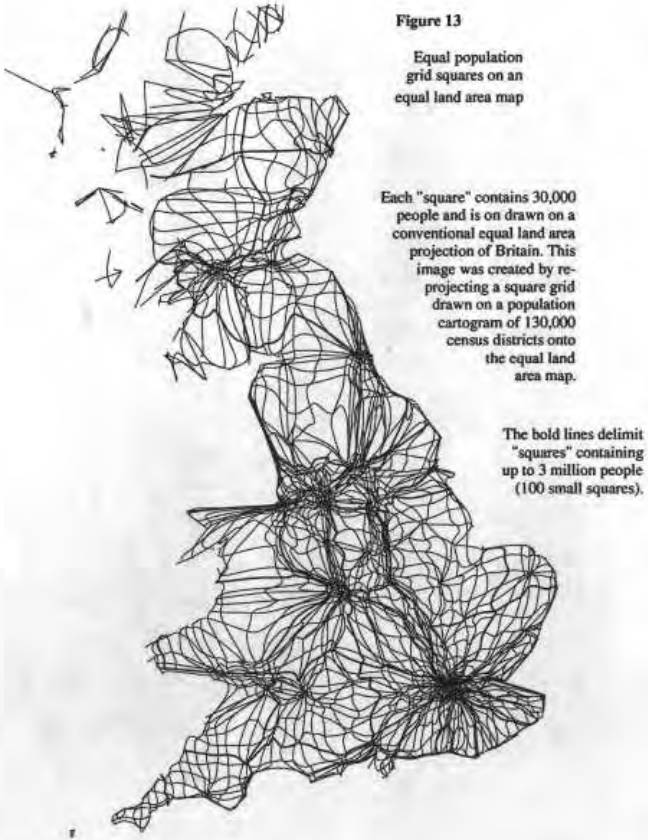


Figure 13

Equal population grid squares on an equal land area map

Each "square" contains 30,000 people and is on drawn on a conventional equal land area projection of Britain. This image was created by re-projecting a square grid drawn on a population cartogram of 130,000 census districts onto the equal land area map.

The bold lines delimit "squares" containing up to 3 million people (100 small squares).

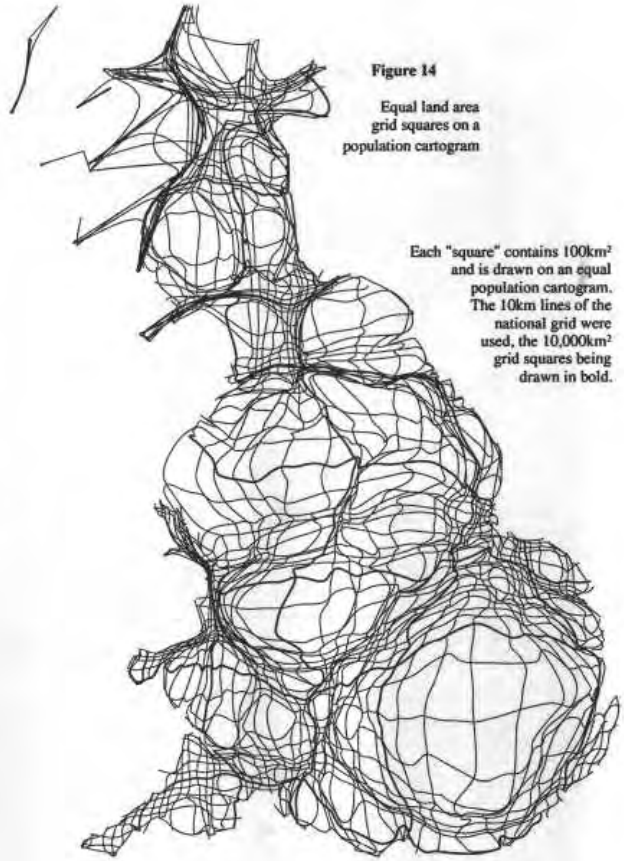


Figure 14

Equal land area grid squares on a population cartogram

Each "square" contains 100km² and is drawn on an equal population cartogram. The 10km lines of the national grid were used, the 10,000km² grid squares being drawn in bold.

Figure 15

Daily commuting flows between English and Welsh wards on an equal land area map

Each individual flow of more than 2% of the employed population of the area of residence is drawn as a thin line to the area of its workplace. The movements of 10,319,230 people are shown, 50% of all commuters.

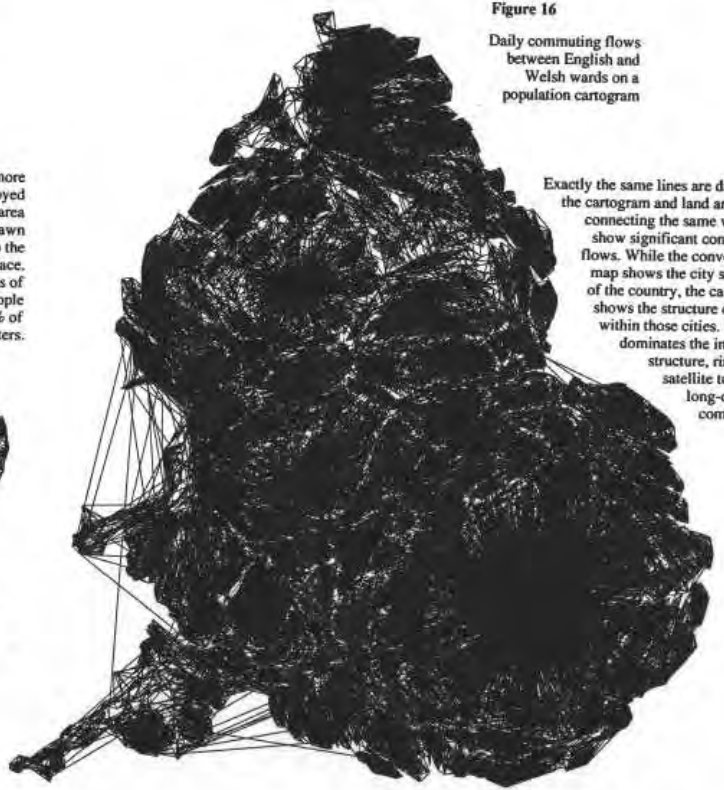


Figure 16

Daily commuting flows between English and Welsh wards on a population cartogram

Exactly the same lines are drawn on the cartogram and land area map, connecting the same wards to show significant commuting flows. While the conventional map shows the city structure of the country, the cartogram shows the structure of flows within those cities. London dominates the industrial structure, ringed by satellite towns of long-distance commuters.

Appendix: Cartogram Algorithm

Being distributed for free academic use only.


```
program cartogram (output);

{Pascal implementation of the cartogram algorithm}

{Expects, as input, a comma-separated-value text
 file giving each zone's number, name, population,
 x and y centroid, the number of neighbouring zones
 and the number and border length of each neighbouring
 zone. Outputs a radius and new centroid for each zone.
 The two recursive procedures and a tree structure are
 include to increase the efficiency of the program.}

{Constants are currently set for the 10,444 1981 census
 wards of Great Britain and for 15,000 iterations of the
 main procedure- exact convergence criteria are unknown.
 Wards do actually converge quite quickly - there
 are no problems with the algorithm's speed -
 it appears to move from O(n^2) to O(n log n)
 until other factors come into play when n
 exceeds about 100,000 zones.}

 const
   iters = 15000;
   zones = 10444;
   ratio = 0.4;
   friction = 0.25;
   pi = 3.141592654;

type
   vector  = array [1..zones] of real;
   index   = array [1..zones] of integer;
   vectors = array [1..zones, 1..21] of real;
   indexes = array [1..zones, 1..21] of integer;
   leaves  = record
               id          : integer;
               xpos        : real;
               ypos        : real;
               left        : integer;
               right       : integer;
   end;
   trees = array [1..zones] of leaves;

var
   infile, outfile            : text;
   list                       : index;
   tree                       : trees;
   widest, dist               : real;
   closest, overlap           : real;
   xrepel, yrepel, xd, yd     : real;
   xattract, yattract         : real;
   displacement               : real;
   atrdst, repdst             : real;
   total_dist                 : real;
   total_radius, scale        : real;
   xtotal, ytotal             : real;
   zone, nb                   : integer;
   other, itter               : integer;
   end_pointer, number        : integer;
   x, y                       : index;
   xvector, yvector           : vector;
   perimeter, people, radius  : vector;
   border                     : vectors;
   nbours                     : index;
   nbour                      : indexes;

  {Recursive procedure to add global variable "zone" to}
  {the "tree" which is used to find nearest neighbours}
  procedure add_point(pointer,axis :integer);
      begin
        if tree[pointer].id = 0 then
          begin
            tree[pointer].id   := zone;
            tree[pointer].left := 0;
            tree[pointer].right:= 0;
            tree[pointer].xpos := x[zone];
            tree[pointer].ypos := y[zone];
          end
        else
          if axis = 1 then
            if x[zone] >= tree[pointer].xpos then
              begin
                if tree[pointer].left = 0 then
                  begin
                    end_pointer := end_pointer +1;
                    tree[pointer].left := end_pointer;
                  end;
                add_point(tree[pointer].left,3-axis);
              end
            else
              begin
                if tree[pointer].right = 0 then
                  begin
                    end_pointer := end_pointer +1;
                    tree[pointer].right := end_pointer;
                  end;
                add_point(tree[pointer].right,3-axis);
              end
          else
            if y[zone] >= tree[pointer].ypos then
              begin
                if tree[pointer].left = 0 then
                  begin
                    end_pointer := end_pointer +1;
                    tree[pointer].left := end_pointer;
                  end;
                add_point(tree[pointer].left,3-axis);
              end
            else
              begin
                if tree[pointer].right = 0 then
                  begin
                    end_pointer := end_pointer +1;
                    tree[pointer].right := end_pointer;
                  end;
                add_point(tree[pointer].right,3-axis);
              end
      end;

{Procedure recursively recovers the "list" of zones}
{within "dist" horizontally or vertically of the "zone",}
{from the "tree". The list length is given by the integer}
{"number". All global variables exist prior to invocation}
 procedure get_point(pointer, axis :integer);
   begin
    if pointer>0 then
     if tree[pointer].id > 0 then
       begin
         if axis = 1 then
           begin
             if x[zone]-dist < tree[pointer].xpos then
               get_point(tree[pointer].right,3-axis);
             if x[zone]+dist >= tree[pointer].xpos then
               get_point(tree[pointer].left,3-axis);
           end;
         if axis = 2 then
           begin
             if y[zone]-dist < tree[pointer].ypos then
               get_point(tree[pointer].right,3-axis);
             if y[zone]+dist >= tree[pointer].ypos then
               get_point(tree[pointer].left,3-axis);
           end;
         if (x[zone]-dist < tree[pointer].xpos)
           and (x[zone]+dist>=tree[pointer].xpos) then
           if (y[zone]-dist < tree[pointer].ypos)
             and(y[zone]+dist)=tree[pointer].ypos) then
             begin
               number := number +1;
               list[number] := tree[pointer].id;
             end;
       end;
   end;
```

```
{The main program}
  begin
    reset(infile,'FILE=ward.in');
    rewrite(outfile,'FILE=ward.out');
    total_dist :=0;
    total_radius := 0;

    for zone := 1 to zones do
      begin
        read(infile,people[zone],x[zone],y[zone],nboura[zone]);
        perimeter[zone] := 0;
        for nb := 1 to nbours[zone] do
          begin
            read(infile,nbour[zone,nb], border[zone,nb]);
            perimeter[zone]:=perimeter[zone]+border[zone,nb];
            if nbour[zone,nb] > 0 then
              if nbour[zone,nb] < zone then
                begin
                  xd := x[zone]- x[nbour[zone,nb]];
                  yd := y[zone]- y[nbour[zone,nb]];
                  total_dist := total_dist + sqrt(xd*xd+yd*yd);
                  total_radius := total_radius +
        sqrt(people[zone]/pi)+sqrt(people[nbour[zone,nb]]/pi);
                end;
          end;
        readln(infile);
      end;
  writeln ('Finished reading in topology');

  scale := total_dist / total_radius;
  widest := 0;
  for zone := 1 to zones do
    begin
      radius[zone] := scale * sqrt(people[zone]/pi);
      if radius[zone] > widest then
        widest := radius[zone];
      xvector[zone] := 0;
      yvector[zone] := 0;
    end;
  writeln ('Scaling by ',scale,' widest is ',widest);

{Main iteration loop of cartogram algorithm.}
  for itter := 1 to iters do
    begin


      for zone := 1 to zones do
        tree[zone].id := 0;
      end_pointer := 1;
      for zone := 1 to zones do
        add_point(1,1);

      displacement := 0.0;

{Loop of independent displacements- could run in parallel.}
      for zone := 1 to zones do
        begin
          xrepel  := 0.0;
          yrepel  := 0.0;
          xattract := 0.0;
          yattract := 0.0;
          closest := widest;
{Retrieve points within widest+radius(zone) of "zone"}
{to "list" which will be of length "number".}
          number := 0;
          dist := widest + radius[zone];
          get_point(1,1);
{Calculate repelling force of overlapping neighbours.}
          if number > 0 then
            for nb := 1 to number do
              begin
                other := list[nb];
                if other <> zone then
                  begin
                    xd := x[zone]-x[other];
                    yd := y[zone]-y[other];
                    dist := sqrt(xd * xd + yd * yd);
                    if dist < closest then
                      closest := dist;
                    overlap:=radius[zone]+radius[other]-dist;
                    if overlap > 0.0 then
                      if dist > 1.0 then
                        begin
                          xrepel:=xrepel-
                                overlap*(x[other]-x[zone])/dist;
                          yrepel:=yrepel-
                                overlap*(y[other]-y[zone])/dist;
                        end;
                  end;
              end;
{Calculate forces of attraction between neighbours.}
          for nb := 1 to nbours[zone] do
            begin
              other := nbour[zone,nb];
              if other <> 0 then
                begin
                  xd := x[zone]-x[other];
                  yd := y[zone]-y[other];
                  dist := sqrt(xd * xd + yd * yd);
                  overlap:=dist-radius[zone]-radius[other];
                  if overlap > 0.0 then
                    begin
                      overlap := overlap*
                            border[zone,nb]/perimeter[zone];
                      xattract:=xattract+
                            overlap*(x[other]-x[zone])/dist;
                      yattract:=yattract+
                            overlap*(y[other]-y[zone])/dist;
                    end;
                end;
            end;
{Calculate the combined effect of attraction and repulsion.}
          atrdst := sqrt(xattract*xattract+yattract*yattract);
          repdst := sqrt(xrepel*xrepel+yrepel*yrepel);
          if repdst > closest then
            begin
              xrepel := closest * xrepel / (repdst + 1);
              yrepel := closest * yrepel / (repdst + 1);
              repdst := closest;
            end;
          if repdst > 0 then
            begin
              xtotal:=(1-ratio)*xrepel+
                    ratio*(repdst*xattract/(atrdst+1));
              ytotal:=(1-ratio)*yrepel+
                    ratio*(repdst*yattract/(atrdst+1));
            end
          else
            begin
              if atrdst > closest then
                begin
                  xattract := closest*xattract/(atrdst+1);
                  yattract := closest*yattract/(atrdst+1);
                end;
              xtotal := xattract;
              ytotal := yattract;
            end;
{Record the vector.}
          xvector[zone]:= friction *(xvector[zone]+xtotal);
          yvector[zone]:= friction *(yvector[zone]+ytotal);
          displacement := displacement+
                    sqrt(xtotal*xtotal+ytotal*ytotal);
        end;

{Update the positions.}
      for zone := 1 to zones do
        begin
          x[zone] := x[zone] + round(xvector[zone]);
          y[zone] := y[zone] + round(yvector[zone]);
        end;
      displacement := displacement / zones;
      writeln('Iter: ', iter, ' disp: ', displacement);
    end;
{Having finished the iterations write out the new file.}
    for zone := 1 to zones do
      writeln(outfile,radius[zone]:9:0,',',x[zone]:9,
                                    ',',y[zone]:9);
  end.
```

MULTIVARIATE REGIONALIZATION: AN APPROACH USING INTERACTIVE STATISTICAL VISUALIZATION

Jonathan R. Hancock
Department of Geography
Syracuse University
343 H.B. Crouse Hall
Syracuse NY, 13244
e-mail: jrhancoc@rodan.syr.edu

ABSTRACT

This paper discusses a new way to use computers in determining geographic regionalizations for the purposes of creating descriptive maps. A program for **computer aided regionalization** is described. Rather than relying on automatic algorithms to delineate regions, this method uses interactive statistical graphics to help the user gain an understanding of data and allows the user to group areas into regions in a trial-and-error fashion. An adaptation of the frame-rectangle symbol, called a "multiple-frame" symbol, is used to communicate the attribute values of areas and the statistical properties of regions. The chief advantages of this approach are that it is flexible, it increases the user's awareness of the data, and it assists in the labeling of regions.

REGIONALIZATION AND DESCRIPTION

Geographic regionalization[1] may be defined as grouping contiguous areas, or more precisely basic spatial units (BSU's), into regions such that *something can be said* about each region. This implies that areas within a region have something in common, such as similar attribute characteristics. In general, regionalization is based on the statistical interpretation of one or more attribute variables from such sources as census, electoral, or property tax data. The method described in this paper, for example, uses the statistical measures of mean and variance and allows the user to consider as many as seven attribute variables.

Regions are delineated for a variety of purposes, and often these purposes require optimizing or standardizing the regions according to well-defined criteria. In election districting, for example, the primary goal is to create regions that are as equal in population and as compact as possible. Regionalizations such as this can be called *functional* because the derived maps have practical importance -- in this case, they determine where people may vote. In contrast, *descriptive* regionalizations are used in making maps that communicate geographical ideas or illustrate spatial

[1]Regionalization differs from classification in that the members of each region must be contiguous.

patterns. Descriptive maps of labeled regions are common in atlases, newspapers, journals, textbooks, and on television; examples are maps of ethnic neighborhoods, political regions, and socio-economic zones. Appropriately, functional regionalizations must be generated according to strict rules; descriptive regionalizations, on the other hand, tend to be more subjective and are not always made in a systematic or scientific fashion. Nonetheless, descriptive regionalizations can be influential and should express geographic facts as accurately as possible. The method introduced in this paper provides a new way of creating such descriptive maps of regions.

Of course, descriptive maps do not need to contain regions -- there are other techniques for conveying spatial information through maps. Examples are classed maps such as choropleth or isarithmic maps, flow maps, and maps of point phenomena. In many instances, however, especially when several attribute variables are considered, a map of labeled regions is the simplest and most effective way to communicate geographic ideas. Maps of labeled regions do not require any knowledge of cartographic techniques and do not require the reader to decipher any cartographic symbols. Also, a map of labeled regions tends to be more memorable and emphatic than other types of maps because it forces the reader to associate shapes (regions) with words (labels).[2] Thus, while maps of labeled regions are less precise and objective than other types of maps, they are easier for the layman to understand.

THE DIFFICULTIES OF REGIONALIZATION

Over the last few decades, geographers have developed a variety of methods for generating regionalizations. The difficulty is that these methods are based on different theories and they yield varied results. Most researchers do not have the opportunity to try different approaches and tend to stick with one method. Often regionalizations are defended on the basis of the method used to create them rather than on the statistical qualities of the derived regions, and researchers seem to have too much faith in the methods they have chosen. Even if a researcher did try different methods, he or she would find it difficult to compare the results and ascertain which method yields the "best" solution, because there is no universally accepted way of evaluating or comparing regionalizations.

Many of regionalization methods use automatic algorithms to try to maximize the homogeneity of regions.[3] . In statistical terms, this is the

[2]For more on the importance of associating verbal descriptions with regions, refer to Rodoman.

[3]There are fundamentally different ways of creating regionalizations including maximizing the difference between regions, maximizing the contrast at borders between regions, and maximizing the spatial autocorrelation of a regionalization.

same as minimizing the variance within regions.[4] The method introduced in this paper also has the goal of maximizing regional homogeneity, but it does not use a solution-finding algorithm. The problem with the algorithmic approaches is that they might miss subtle but important patterns in the data or might emphasize the wrong criteria in judging regions. Also, there may be additional, perhaps even non-quantifiable, factors that the regionalizer wishes to consider in setting up his or her regionalization.[5] An automatic algorithm is not equipped to consider factors that are not clearly defined and numerically expressed. In short, an automatic algorithm may be too inflexible.

A NEW APPROACH TO REGIONALIZATION

I propose a non-algorithmic approach to regionalization which takes advantage of the computer to speed up and facilitate the delineation of regions but does not actually determine regions. I call this approach **computer aided regionalization**, and it is the basis of a program I am developing named *Look 'n Link*.[6] It is important to emphasize that computer aided regionalization is *not* automatic; it is an environment that enables the user to delineate regions in a trial-and-error fashion by providing interactive feedback concerning the quality (i.e., homogeneity) of regions as they are being created.

This regionalization method is open ended -- the process is complete when the user is satisfied with the quality of the regionalization and the degree of aggregation. The number of regions, the degree of homogeneity, and the amount of time spent deriving regions are entirely up to the user's discretion. As a result, computer aided regionalization cannot guarantee that the regions will be homogeneous or compact,[7] and it could be used to create bad regionalizations. Regionalizations made in this manner cannot be defended on the basis of how they were generated; they can only be defended on the basis of the statistical properties of the results.

With *Look 'n Link*, the user builds regions by repeatedly joining adjacent areas or regions. The program starts off by displaying a base map showing the areal unit boundaries. Centered over each area is a multiple-variable graphic symbol, called a **multi-frame** (described below), that indicates the values of several variables for each area. After viewing these

---

[4]Variance expresses how loosely the values of constituent areal units are dispersed about a region's mean (average) value, and hence describes the degree of heterogeneity within a region; it is defined as "the average squared difference between an observed value and the arithmetic mean." (Griffith and Amrhein, 1991)

[5]Examples of these non-quantifiable factors are personal impressions of regional patterns, traditionally accepted affiliations among areas, and major physical barriers.

[6]I call the program *Look 'n Link* because it allows the user to *look* at the map to gain an understanding of the data and then *link* areas together to create regions. The program is written in Think Pascal and runs on the Macintosh.

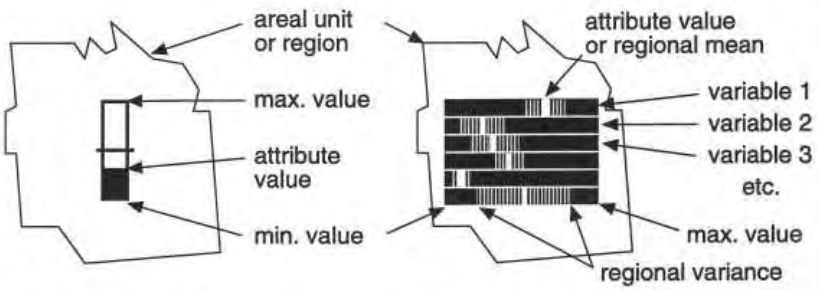
[7]It *does* guarantee that the regions will be contiguous.

symbols, the user selects a pair of adjacent areas to join and clicks on the boundary between these areas to "link" them. The two areas' multi-frames are replaced by a single multi-frame representing the combined region and indicating the values and variances for that region. Each time the user adds an area to a region he can see how that addition effects the region's homogeneity. The user can also remove an area from a region, so he or she is free to try out different combinations of areas.

THE MULTI-FRAME SYMBOL

I had to invent a way of displaying multivariate data so that it is easy to compare neighboring areas based on a set of variables. I rejected the idea of using multiple maps because this would force the user to switch back and forth between different views; I therefore needed some kind of multiple-variable point symbol. I considered using Chernoff's faces,[8] but I felt that the way these symbols present variables (i.e. by the shapes of different physiognomic features) could distort the results because some people might notice noses more than eyes. I tried star symbols (glyphs),[9] but when combined with a base map these created an overload of directional information. The best solution seemed to be an adaptation of the frame rectangle symbol.[10]

The operative metaphor for a frame rectangle symbol is that of a thermometer: the height of the "fluid" corresponds to the associated attribute value. Place several frame rectangles together and turn them on their side, and you have the **multi-frame symbol**.



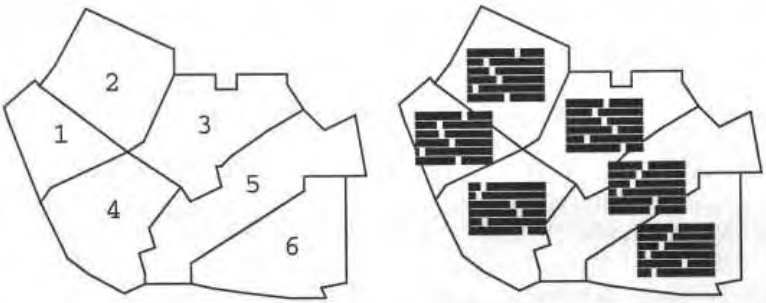
Frame-rectangle symbol "Multi-frame" symbol

Each of the horizontal bars in the multi-frame symbol corresponds to a particular attribute variable. The location of the vertical white stripe on a horizontal bar indicates the attribute value of an area (BSU) or the mean attribute value of a region (group of BSU's). If the symbol represents a

[8]See Feinberg, p. 173.
[9]See Dunn and Walker, p. 1400.
[10]The frame-rectangle symbol has been used by Mark Monmonier.

region, there may also be a shaded area around the white stripe, the size of which indicates the variance of the region with respect to that variable. This shaded zone can be thought of as a representation of the "fuzziness" of a region.

The different horizontal bars of the multi-frame are color coded to help the user associate them with their respective variables. The user can change the order and number of the variables represented, but all multi-frames will display the same set of variables. In comparing multi-frame symbols, the user need not think about individual attribute values. Instead, he or she can focus on the collective configuration of white stripes, which I refer to as the **profile** of a region or area. The profile visually summarizes a region's attribute characteristics. To select areas for joining, the user scans the map for neighboring areas with similar profiles.

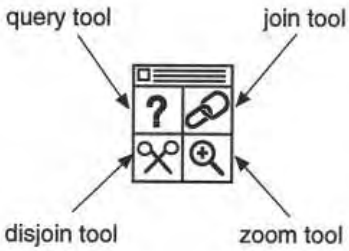


In the hypothetical example above, the user might start by joining areas 1 and 2 or areas 5 and 6 because these pairs of regions have the most similar profiles.[11] Areas 1 and 2 seem to be the closest match. Areas 5 and 6 have significantly different values for the last variable, but are strikingly similar in terms of the other variables. The multi-frames for areas 3 and 4 have some similarities, especially with regard to the last four variables.

CREATING REGIONS

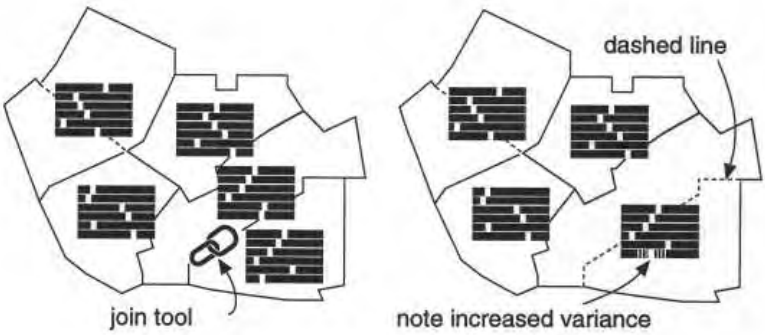
Like other Macintosh applications, *Look 'n Link* features a graphic menu or "toolbox" that allows the user to select one of several modes of operation or "tools". When a tool is selected, the appearance of the cursor changes accordingly. The chosen tool is positioned with the mouse and activated by pressing the mouse button. The use of **zoom tool** (magnifying glass) is obvious -- it adjusts the scale of the displayed map so that the multi-frame symbols (which remain constant in size) do not obscure each other or the areal boundaries.

[11]These multi-frame symbols do not indicate any variance because they represent BSU's, not regions.



Look 'n Link toolbox

The **query tool** (question mark) is used to access information about the map. When a user clicks on an area or region, a balloon appears containing information such as the names and populations of the areal unit(s). By clicking within a multi-frame symbol, the user can see a display of a variable's name, numerical expressions of the regional mean and variance, and the name and value of the weighting variable (if appropriate). For more detailed information on regions, the user may print out statistical summaries that list the regional means, variances, minimums, and maximums for all variables.



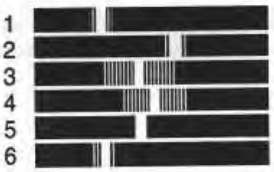
Regions are created with the join and disjoin tools. The **join tool** (chain links) is for connecting two adjacent areas to create a region, which the user does by clicking on the boundary between the areas. The boundary changes from a solid line to a dashed line and the two areas' multi-frames are replaced by a single multi-frame for the composite region. The figure above shows what happens when areas 5 and 6 are joined; note that the multi-frame indicates an increase in variance for the last variable. Only adjacent (contiguous) areas may be linked; this limitation is necessary to guarantee that regions remain contiguous.[12] One can undo a join between two areas by clicking the **disjoin tool** (scissors) on a dashed boundary.

[12]While most thematic mapping packages store boundary information in a non-topological ("spaghetti") format, *Look 'n Link* requires the more complicated topological (point-line-polygon) data model because the program uses chain processing to determine the areas or regions along a boundary, to alter the appearance of boundary segments, and to check the contiguity of areas.

LABELING THE REGIONS

While the program described here could be used merely as a data exploration tool, it is intended to help people create presentation maps, although it will not actually print them out[13]. The user is discouraged from using the multi-frame symbol on a static map; it would be too hard for most audiences to decipher. Instead, the map maker should translate these symbols into concise and understandable *labels*. Writing labels is far from trivial, and may be the most crucial step in creating a good map of regions. To determine these labels, the map maker must study the multi-frames and the statistical summaries for each region.

Var. ID	Variable Name	Regional Mean	Regional Variance
1	%Mondale84	28.2	8.3
2	%Reagan84	62.1	9.7
3	%Dukakis88	46.5	25.1
4	%Bush88	53.2	25.0
5	%Clinton92	55.5	6.7
6	%Regan92	31.2	11.1



In this hypothetical example, a researcher is interested in political regions, or more specifically, regionalization based on voting in the last three presidential elections. Illustrated here are the statistical summary and multi-frame for one particular region. This region was strongly pro-Reagan in 1984 and was pro-Clinton in 1992. The variances for the 1984 and 1992 election results are relatively low, implying that this region is fairly homogeneous in terms of the voting patterns for these years. The situation for 1988 is different. Not only is Reagan's margin of victory slimmer, but the variances are much higher. This means that the region is not as consistent with regard to 1988 voting patterns. In describing this region, the user should focus on those years for which the variances are low, ignoring the year with high variances. Thus, he or she might label the region "REAGAN IN '84, CLINTON IN '92." Other regions on the same map might be labeled, "CONSISTENTLY REPUBLICAN," "REAGAN REPUBLICAN," or "DUKAKIS IN '88." To justify his or her selection of regions and regional labels, the user could append to the map detailed summaries of the regional statistics .

COMMENTS

Some might criticize computer aided regionalization on the basis that it is too "subjective." R.J. Johnston demonstrated, however, that the so-called "objective" approaches to classification are actually subjective, mainly because the results of regionalization depend largely on the choice of the

[13]The program will allow users to export regional boundary data to an illustration package, such as FreeHand, wherein labels, legends, titles, etc. may be added.

method used (Johnston, 1968). Many of these objective methods themselves have subjective aspects, such as the choice of indices or factors, the choice of a cut-off level in hierarchical grouping, or the choice of significance levels in methods involving inferential statistics. Most regionalization algorithms fail to adequately deal with these unintended subjective influences. In contrast, computer aided regionalization welcomes subjective influences. The way the user creates regions (i.e., based on perception) is subjective; on the other hand, the way the computer creates the multi-frame symbols (i.e., using statistical measures) is objective. Computer aided regionalization is thus a compromise between an objective and a subjective approach.

A legitimate criticism of computer aided regionalization, as well as other approaches to regionalization, concerns the aggregation problem.[14] Several geographers, most notably, S. Openshaw, have written about the aggregation problem, also known as the modifiable areal unit problem (MAUP).[15] In a 1991 article, A.S. Fotheringham and D.W.S. Wong wrote that,

> The modifiable areal unit problem is shown to be essentially unpredictable in its intensity and effects in multivariate statistical analysis and is therefore a much greater problem than in univariate or bivariate analysis. The results of this analysis are rather depressing in that they provide strong evidence of the unreliability of any multivariate analysis undertaken with data from areal units. (Fotheringham and Wong, 1991, p. 1025)

My attitude about the aggregation problem is that it *is* rather depressing, but, like a lot of things in life, you just have to accept it and go on. The only consolation I can offer is that at least computer aided regionalization does not presume to be entirely objective, so it cannot be said that MAUP spoils an otherwise "valid" result. On the other hand, it is essential that the user be aware of the aggregation problem and that he or she qualify his interpretation and acknowledge the limitations that are due to the given set of areal units.[16]

Some might say that computer aided regionalization, because of its inherent flexibility, could make it easier for ill-intentioned individuals to create gerrymandered or otherwise deceptive maps. My response is that, if people want to lie or cheat with maps, they'll find a way to do it whatever software they're using. A computer program cannot be a policeman! *Look 'n Link* does indicate the reasonableness of a regionalization, so at least one could not make a bad map without being aware of it. In contrast,

---

[14]For most applications, data is available only for a predefined set of enumeration districts (such as census tracts or counties) because the census bureau or other agency, for reasons of confidentiality, does not release individual level data. Therefore, while it is possible to know the total or mean values for each BSU, it is impossible to know how those values are distributed, spatially or otherwise, within that BSU.

[15]See Openshaw, 1981.

[16]I thank Mark Monmonier for this insight.

many of the automatic approaches provide no mechanism for verifying how good the results are. Computer aided regionalization encourages a healthy awareness of the quality of the results, and such awareness should engender honesty. I might also respond that automatic regionalization algorithms may be unethical because they defer interpretation to a machine. Are we to let computers define our world and describe our society? Computers can be wonderful tools, but they should not determine how we understand ourselves.

Whatever method is used -- automatic or non-automatic -- a regionalization necessarily involves some loss of information. It is the responsibility of the map maker to see that this loss of information is not harmful. Nonetheless, any regionalization, whether automatically derived or not, should be critically evaluated and viewed with an appropriate measure of skepticism.

REFERENCES

Dunn, R., and R. Walker. 1989. District-level variations in the configuration of service provision in England: A graphical approach to classification. *Environment and Planning, A* 21:1397-1411.

Feinberg S.E. 1979. Graphical methods in statistics. *The American Statistician* 33:165-178.

Fotheringham A.S., and D.W.S. Wong. 1991. The modifiable areal unit problem in multivariate statistical analysis. *Environment and Planning, A* 23:1025-1044.

Gilbert, E.W. 1960. The idea of the region. *Geography* 45:157-75.

Griffith, Daniel A., and C.G. Amrhein. 1991. *Statistical Analysis for Geographers*. Englewood Cliffs, NJ: Prentice Hall.

Harley, J. B. 1991. Can there be a cartographic ethics? *Cartographic Perspectives* 10:9-16.

Johnston, R.J. 1968. Choice in classification: The subjectivity of objective methods. *Annals of the Association of American Geographers* 58:575-589.

MacEachren, A.M. 1991. The role of maps in spatial knowledge acquisition. *The Cartographic Journal* 28:152-162.

MacEachren, A.M., and M. Monmonier. 1992. Introduction. *Cartography and Geographic Information Systems* 19(4):197-200.

Monmonier, Mark 1991. *How to Lie with Maps.* Chicago: University of Chicago Press.

Openshaw, S. 1977. A geographical solution to the scale and aggregation problems in region building, partitioning and spatial modeling. *Transactions, Institute of British Geographers* n.s. 2:459-72.

Openshaw, S. 1984. Ecological fallacies and the analysis of areal census data. *Environment and Planning, A* 16:17-31.

Openshaw, S., and P.J. Taylor. 1981. The modifiable areal unit problem. In *Quantitative Geography: A British View*, N. Wrigley and R.J.Bennet, eds., Boston: Routledge & Kegan Paul.

Rodoman, B.B. 1967. Mathematical aspects of the formalization of regional geographic characteristics. *Soviet Geography* 8:687-708.

VISUALIZATION OF INTERPOLATION ACCURACY

William Mackaness mack@olympus.umesve.maine.edu
Kate Beard beard@olympus.umesve.maine.edu
NCGIA
348 Boardman Hall
University of Maine
Orono, Me 04469

ABSTRACT

Spatial interpolation is a valuable and now more frequently used function within GIS. A number of data sources for GIS include interpolated values (e.g. DEM's) and a number of the output products from GIS are the results of some form of interpolation. Since interpolated results in one way or another play a role in GIS analysis, users can benefit from some assessment of the reliability of interpolated values. The accuracy of an interpolation depends on several factors including: sampling scheme, number of sample points, interpolation method, measurement error in the observed x, y, z values and the nature and complexity of the observed phenomena.

Most interpolation methods provide no information on the reliability of the estimated values. Kriging is one exception which produces estimates of values at unrecorded places without bias and with minimum and known variance. Results of kriging reported in the literature typically show kriged values and error estimates as separate isarithmic maps making assessment of the results difficult. This paper describes visualization techniques for examining the reliability of interpolated values. Visual displays are presented which account for the number and distribution of sampling points, the interpolation method, and measurement error in observed values. Additionally, the paper addresses displays for kriged results which combine kriged values and error estimates.

INTRODUCTION

Ideally a GIS should indicate graphically and numerically the reliability of each analysis. Spatial interpolation defined as a process for estimating values at unsampled locations from a set of sample points should be no exception. Results of an interpolation function are commonly displayed as isolines. Lacking other evidence the assumption is that these isolines and the underlying interpolated values are uniformly reliable over the interpolated area. If we have knowledge that this is not the case, then the differential uncertainty in the results should be communicated. Users should be able to ascertain which areas are least reliable and where possible be provided with information to understand why these values are less reliable.

Interpretations of reliability may utilize 'several imaging functions to display the data in different ways and to provide complementary information' (Farrell, 1987, 175). With respect to interpolation, one can envisage a progression of graphical representations to convey variability and reliability in the resulting values. At the simplest level, graphical representation of the location of sample points provides clues to spatial variation in the reliability of interpolated values. Regions of sparse data are unlikely to produce

accurate interpolations irrespective of technique. Indeed several interpolation methods depend substantially on the selection of appropriate neighbors (Gold 1989). Thus conveying the spatial distribution of measured points is an important indicator of reliability. Traditionally interpolation functions have treated sample points as having uniform positional accuracy and uniformly accurate z values. Simple displays of sample point locations can be extended through symbology to indicate variation in positonal or attribute accuracy of the sampled points. The information on positional and attribute accuracy can arise through adjustment analysis (e.g. the creation of error ellipses - Hintz and Onsrud 1990), or through comparison of data collected using different technology or degrees of accuracy.

Finally, display may be designed to communicate information on the actual interpolation method. The various interpolation methods themselves can generate varying patterns of reliability in the results. Interpolation methods (see reviews by Lam 1983, Burrough 1986, Ripley 1981) produce different spatial patterns of reliability through their underlying assumptions and parameters. The next sections of the paper presents a suite of displays to communicate the quality of interpolations across the different levels just summarized.

DISPLAYS OF SAMPLED POINTS

Since the number and distribution of sampling points contribute substantially to the interpolation outcome, a simple but effective visual assessment tool is a capability to toggle on and off the sample point locations (see Figure 1).

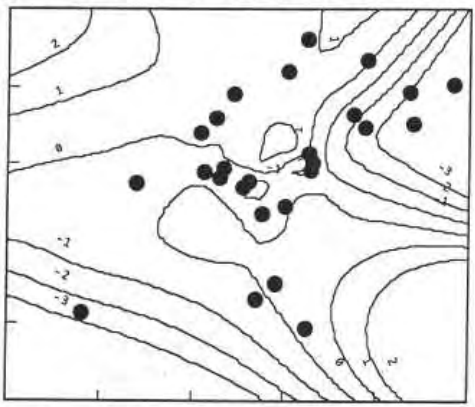


Figure 1. *Sample points locations displayed with isolines.*

As an extension to this display, points may be classified and color coded by their Z values. A similar color coding of isolines can then clearly indicate deviations from measured values as well as indicate the areas of sparse data. Figure 2 illustrates such a display using gray values.

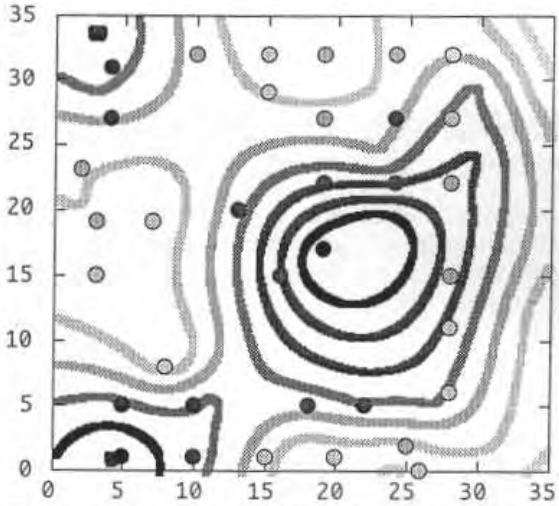


Figure 2. *Gray scale version of sample points values displayed with isolines.*

When data sets of mixed heritage are combined to form the point set for interpolation, displays may be designed to document variations in accuracy among these points. For example, water quality samples taken over a ten year time period may have different accuracies due to the sampling technique, laboratory quality control procedures, instrument calibration or other factors. In this case the sample points may be color coded or otherwise symbolized by an accuracy measure. From this sort of display, spatial clusters of sample points with lower accuracies may be identified and hence locations where interpolated results may be less reliable. Variation in the positional certainty of sample points may be displayed in a similar fashion.

DISPLAYS OF INTERPOLATION METHOD RELIABILITY

A variety of interpolation methods have been documented in the literature. By understanding the behavior of these methods or through the actual computation of these methods, error estimates may be generated and subsequently displayed. Table 1 extracted from Burrough (1986) summarizes these methods.

Method	Deterministic/Stochastic	Local/Global	Exact Interpolator
Proximal	deterministic	global	no
Trend surface	stochastic	global	no
Fourier Series	stochastic	global	no
B-splines	deterministic	local	yes
Moving average	deterministic	local	no
Optimal (kriging)	stochastic	local	yes

Table 1 *Interpolation methods summarized from Burrough 1986.*

Visual displays which illustrate the spatial reliability of interpolated results are developed for two of these methods: weighted moving average and kriging.

WEIGHTED MOVING AVERAGE

This method uses a set of data points in proximity to an interpolation location and performs an averaging process. The averaging process includes weights proportional to the distance of the data point from the estimation point. The important parameters which need to be specified for this method are the size of the neighborhood and the appropriate weighting function. Both of these factors can have significant effects on interpolation results. The other important factor is the distribution of sample points with respect to the estimation point as results are very sensitive to clustering in the sample points. (Ripley 1981). One visualization method can not capture all these influences, but a technique which captures the distance factor is displayed in Figure 3. In this figure, distances were computed from sampling points and displayed as a gray shade image. When displayed as a backdrop to isolines, darker areas indicate areas which may be less reliable.

Figure 3. *Shading is based on a distance function computed from sample points. Moving weighted average interpolated values in darker areas are less reliable.*

KRIGING

There are numerous papers that cover the topic of kriging in various levels of detail (for example Oliver and Webster 1986, Dunlap and Spinazola 1984; Doctor 1979; Cressie and Hawkins 1980; Bridges 1985; Ripley 1981; McBratney and Webster 1986; Royle et al. 1981). Kriging is an interpolation technique that generates an estimated surface from a regular or irregular set of points. The principle advantage of Kriging over the many other interpolation techniques in existence is that it provides an indication of the error associated with interpolated values and is the only method that uses statistical theory to optimize the interpolation (Clarke 1990). Kriging has been successfully used in the spatial prediction of soil properties (Burgess and Webster 1980), mineral resources, aquifer interpolation (Doctor 1979; Dunlap and Spinazola 1984), soil salinity through interpolation of electrical conductivity measurements (Oliver and Webster 1990), meteorology and forestry.

Deterministic models assume that we know a great deal about the behavior of the variable which is rarely the case in the geographical sciences. In contrast to deterministic models, Kriging makes no attempt to describe the physical mechanism of the underlying phenomenon. The advantage of Kriging over polynomial fitting procedures (such as Trend Surface Analysis– TSA) is that Kriging uses a correlation structure among the observations and is more stable over sparsely sampled areas whereas estimates in TSA are greatly affected by the location of data points and can produce extreme fluctuations in sparse areas.

Kriging is based on the regionalized variable theory (Matheron 1971) that accounts for spatial variation in three parts: an underlying structure, a random but spatially correlated component and noise. The regionalized variable theory assumes the spatial variation in the phenomenon represented by the z value is statistically homogenous throughout the surface – the same pattern of variation can be observed across the entire surface. Thus datasets known to have pits and spikes or abrupt changes are not appropriate for use with Kriging.

Kriging requires a number of steps; first the underlying structure is estimated using the semi–variogram.

THE SEMI-VARIOGRAM

Kriging requires that we first compute the semi–variogram, and this is then used to determine the weights and search extent when predicting a value at an unrecorded place. The semi–variogram provides information of the form of the relationship between two observations as a function of intervening distance. The semi–variogram is a graph of the variability of the difference of the regionalized data versus distance between points (known as the lag). The semi–variogram is approximated from the given set of measured values of a regionalized variable. A variety of functions can be used to model the semi–variance and there are a number of theoretical semi–variograms (for discussion see McBratney and Webster 1986). The three most commonly used are the linear, spherical and exponential models.

The regionalized variable is *isotropic* when the semi–variograms is a function only of distance between data points. To determine whether there is any appreciable change in variance with direction, it is common to generate and compare the variograms of two and sometimes eight cardinal directions. Thus kriging is an exploratory technique requiring some experimentation. As McBratney and Webster observe, 'choosing an appropriate semi-variogram is still something of a mystery' (McBratney and Webster 1986, 618) selecting an appropriate model requires some trial and error and thus requires a good understanding of the ontology of the phenomenon under interpolation.

DISPLAYING KRIGING ERRORS OF ESTIMATE

Kriging gives a variance estimate at each interpolated point (called kriging error or errors of estimate). They define confidence intervals about the interpolated points if the errors are assumed to be normally distributed, and kriging errors tend to be greatest in areas of sparse data. Generating the error of estimates helps identify areas in need of further sampling and can thus

improve the quality of sampling in a very specific way (Wright 1983; Keith 1988).

In the following figures error estimates are generated from a series of seismic profiles recorded from a ship within the Gulf of Maine in the Casco Bay area (Kelley et al. 1987) . The z values were digitized from these seismic profiles; three 'layers' were identified – the bedrock, the Holocene and the glaciomarine. In processing the data, the first stage was to find the semi–variogram model that best fit the data. The exponential model was found to be the best fit for bedrock and Holocene layers, and Gaussian for the Glaciomarine. Figure 4 shows two semi-variograms; the Gaussian semi–variogram for Glaciomarine and Exponential for the bedrock and island data set (both generated using Arc/Info*).

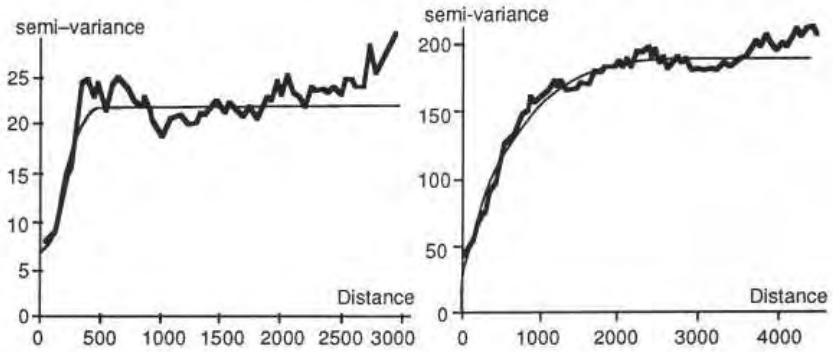


Figure 4. *Experimenting to find the best model of the semi–variogram.*

The variogram was used to generate a regular matrix of both the interpolated values and the error estimates associated with each value (using SpyGlass Transform*). A contour module within Transform generated the isopleth maps and a inverted greyscale was used to create the variance map. In this map, light areas are regions of low variance and dark patches are conversely high. The estimation variance, when overlayed with the resulting contours (as in Figure 5) reveals areas of poor control and is an important factor in the rejection of unreliable edge points. This display assimilates both variables, and has the advantage that the dark regions conceal from view those areas for which the interpolated values are less reliable.

* Mention of software products is not an endorsement of their use.

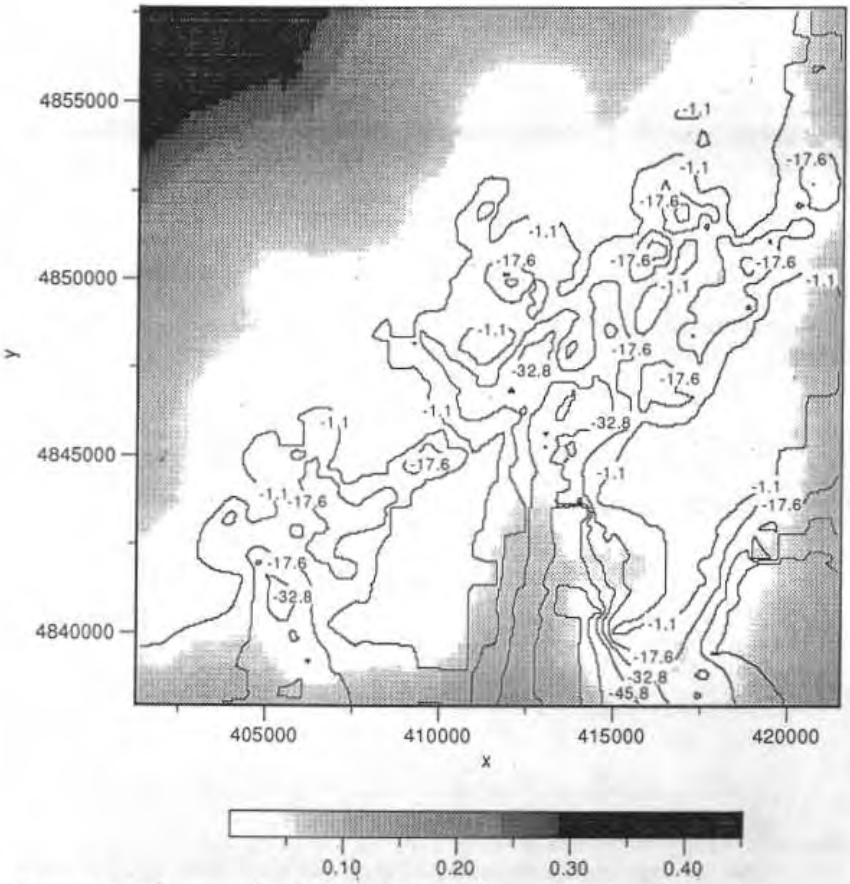


Figure 5. *Contour map of Bedrock and Island bathymetry with variance values overlayed in greyscale.*

Though the variance matrix can be toggled off so as to reveal the results in areas of high variance, an alternative approach is to combine the two variables such that the error of estimates is conveyed either through line fill pattern or thickness of line. Thus the isolines themselves carry the information on reliability. This approach is illustrated in Figure 6. This method summarizes the results of kriging from Oliver and Webster (1990). In this figure the lighter fuzzier portions of the isolines correspond to areas with higher error estimates. Using either method it is possible to incorporate additional information on the map.

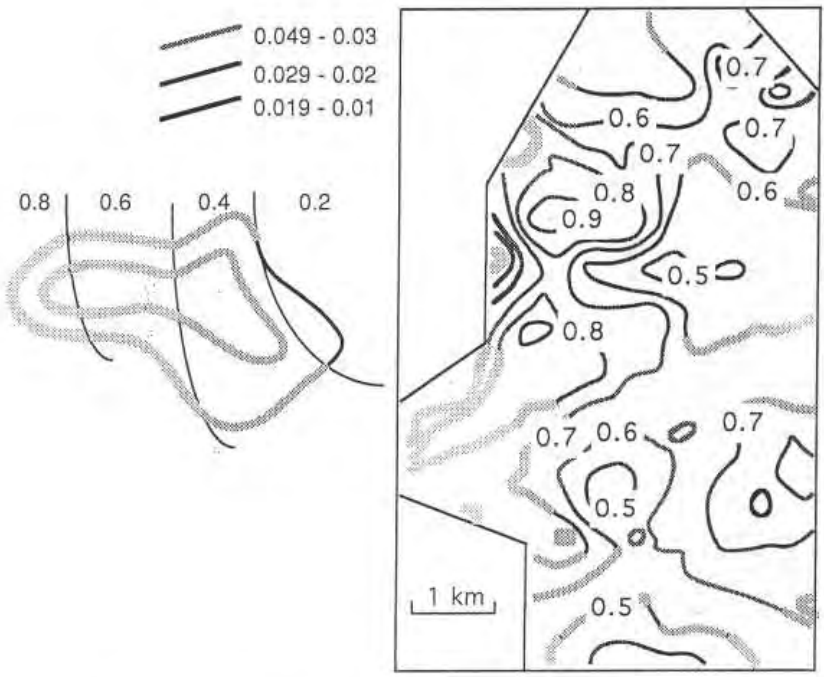


Figure 6. *Combining semi variance value with kriged values through the use of line shading.*

CONCLUSION

The integrity and general worth of any type of analysis is dependent on several factors. These include the quality of first abstraction (sampling regime, resolution of sampling), data manipulation prior to storage, subsequent abstraction (for example digitizing or subsampling), techniques used in the analysis (and degree with which parametric requirements are met), recording of error and subsequent visualization of each of these steps. Thus the concept of 'quality' pervades the entire research design and an overall picture requires knowledge of error and potential variation associated with each stage of processing. This paper has focused on one component of the 'quality chain', namely degree of certainty associated with interpolation.

This paper has illustrated a sampling of techniques that in combination may be used to display the reliability of interpolated values. The methods range from displaying sample point locations, values, and accuracy, to displaying the variation in reliability which is produced as a function of the interpolation method itself. Kriging has the advantage of generating error estimates as the result of the process. However, few tools have been available to view the interpolated values in combination with the error estimates. Two techniques are illustrated to view estimated values and their reliability simultaneously. This represents a small sample of possibilities, yet examples which could be easily incorporated within systems to assist users in their decision making.

ACKNOWLEDGMENT

Both authors are grateful for support from the NSF for the NCGIA under grant No. SES 88-10917. Our thanks to Lori Dolby for her analysis and comment.

REFERENCES

Bridges, N.J. 1985. Kriging: An interactive program to determine the best linear unbiased estimation. US Dept. of Interior, Geological Survey.

Burgess, T.M. and R. Webster. 1980. Optimal interpolation and isarithmic mapping of soil properties II. Block Kriging. Journal of Soil Science Vol. 31, pp. 333.

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment, (Monographs on Soil and Resources Survey No.12), Oxford University Press, New York.

Clarke, K. C. 1990. Analytical and Computer Cartography. Englewood Cliffs, New Jersey: Prentice Hall.

Cressie, N. and D.M. Hawkins. 1980. Robust estimation of the variogram I. Mathematical Geology Vol. 12, pp. 115.

Deutsch C.V. and Journel A.G. 1992. GSLIB Geostatistical Software Library and Users Guide New York:Oxford University Press.

Doctor, P.G. 1979. An Evaluation of Kriging Techniques for High Level Radioactive Waste Repository Site Characterization. Pacific Northwest Lab, Richland, Washington 99352:

Dunlap, L.E. and J.M. Spinazola. 1984. Interpolating Water Table Altitudes in West-Central Kansas Using Kriging Techniques. United States Gov Printing Office Washington.

Farrell, E.J. 1987. Visual Interpretation of complex data IBM Systems Journal Vol. 26, No. 2, pp. 174 - 199.

Gold C.M. 1989. Surface Interpolation, Spatial Adjacency and GIS in J. Raper(ed) Three Dimensional Applications in Geographical Information Systems. London: Taylor and Francis. pp. 21-36.

Haining, R. 1990. Spatial data analysis in the social and environmental sciences Cambridge: Cambridge University Press.

Hintz, R.J. and H.J. Onsrud. 1990.Upgrading Real Property Boundary Information in a GIS URISA Journal Vol. 2, No. 1, pp. 2-10.

Isaaks, E.H. and Srivastava, R.M. 1989. An introduction to Applied Geostatistics New York:Oxford University Press.

Lam, N. S. 1983. Spatial Interpolation Methods: a Review. American Cartogrpaher. 10. 129-49.

Keith L.H. 1988. Principles of Environmental Sampling ACS Professional Reference Book, American Chemical Society.

Kelley, J.T., Belknap, D.F., and Shipp, R.C. 1987. Geomorphology and Sedimentary Framework of the Inner Continental Shelf of South Central Maine. Department of Conversation. Maine Geological Survey, Open File Report No. 87-19.

Krige, D.G. 1966. Two dimensional weighted moving average trend surfaces for ore-evaluation Journal of the South African Institution of Mining and Metallurgy Vol. 66, pp. 13 - 38.

Leenaers, H., P.A. Burrough and J.P. Okx. 1989 Efficient mapping of heavy metal pollution on floodplains by co-kriging from elevation data. in J. Raper(ed) Three Dimensional Applications in Geographical Information Systems. London: Taylor and Francis. pp. 37-51.

Matheron G. 1971. The Theory of Regionalized Variables Les Cahiers du Centre de Morphologie Mathematique de Fontainbleu, 5, pp. 1-210. Ecole Nationale Superieure des Mines de Paris.

McBratney, A. B., and Webster, R. 1986. Choosing Functions for Semi–variograms of soil properties and fitting them to sampling estimates. Journal of Soil Science, vol. 37, pp. 617-639.

Olea, R. (ed) Geostatistical Glossary and Multilingual Dictionary. Oxford University Press New York.

Oliver, M.A. and R. Webster. 1986. Combining nested and linear sampling for determining the scale and form of spatial variation of regionalised variables. Geographical Analysis Vol. 18, pp. 227.

Oliver, M.A. and R. Webster. 1990. Kriging: A method of interpolation for geographic Information systems. International Journal of Geographic Information Systems Vol. 4, No. 3, pp. 313 - 332.

Ripley B. D. 1981 Spatial Statistics New York: John Wiley

Royle, A. G.,Clausen, F. L., and Frederiksen, P. 1981. Practical Universal Kriging and Automatic Contouring. Geoprocessing, Vol. 1, pp. 377-394.

Wright T. 1983. Statistical Methods and the Improvement of Data Quality. Academic Press, Orlando Florida.

VISUALIZING GEOGRAPHIC DATA THROUGH ANIMATION

Donna Okazaki
Department of Geography
University of Hawaii at Manoa
Honolulu, HI 96822

ABSTRACT

Visualization* of spatio-temporal data is enhanced through the use of various computer tools which make it possible to create realistically rendered, 3-D animations. This paper and video presentation will show the process and results of a cartographic animation project, which transformed paper maps and statistical tables into a presentation quality, 3-D animated map. Viewer reactions to the animation have been very positive, resulting in continuing effort in this area.

INTRODUCTION

Cartography is being continually impacted by advances in computing. This is particularly true in the case of geographic data visualization where technology has allowed data representations to evolve from static, printed maps to dynamic, interactive, graphic animations. High level software tools such as Khoros** (Rasure and Young 1992) allow cross-platform synthesis to take place, combining the strengths of different systems into a streamlined computational process. This makes it possible to create multiple perspectives of data which are vital to the scientific inquiry process.

The goal of this project was to create a thematic animation of Japan census data with the objective of being able to view and explore the urban development of Tokyo in time-space. Due to limitations of computer hardware and software, it was not possible to create an interactive, real-time, on-screen animation, so the results were transferred to videotape. However, it is important to stress the interactive nature of the animation process itself which employs visualization in many intermediate steps.

VISUALIZATION THROUGH ANIMATION

The main role of visualization in data analysis is to allow the researcher to interact with data representations during the investigative stage where data are explored to extract information or confirm results (Springmeyer, et. al. 1992). MacEachren and Ganter (1990) state that "The goal of cartographic visualization ... is to produce scientific insights by facilitating the identification of patterns, relationships, and anomalies in data." Animation is viewed as an effective tool

---

* In this paper, the term "visualization" is used exclusively in the context of research methods as a means of data exploration.

**An open software product available through anonymous ftp from ftp.eece.unm.edu or by requesting an order form via email to khoros-request@chama.eece.unm.edu.

for visualizing complex phenomena which can be mapped into time-dependent 2-D or 3-D data representations (DiBiase, et. al. 1992; Gershon 1992; Sparr, et. al. 1993). However, the technology to support animation poses challenges which are related to the availability of fast computer hardware, the resolution supported by video output devices and the state of currently available software.

The first challenge, inadequate computing power, is a common one. When available processor speed is insufficient to produce real-time, animated computer displays, videotaped animations are a cost-effective means of creating these unique data views. Animations of large amounts of data which result from complex computations are especially amenable to this low-cost approach (Choudry 1992). This allows visualization of numerically intensive simulation models or in the case of this project, allows the visualization of complexly rendered 3-D views of data.

The human visual system compensates for the second challenge. There is evidence to support the idea that animation enhances the visibility of features embedded in data. Gershon (1992) claims, "It has been found that observing motion triggers a set of fast and effective mechanisms in the visual system that allow the efficient detection of moving features." This would compensate for the loss of resolution resulting from transferring high quality computer images to consumer-grade videotape.

The third challenge is the most difficult. General purpose visualization packages such as Khoros (Khoral Research, Inc.), AVS (AVS, Inc.) and Explorer (SGI, Inc.) are sufficient for data viewing up to the computational limits of the system, but do not have many of the features of a complete animation system. They also aren't designed to capture and store geographic information. However, these packages are extensible, so they can be used to bridge between GIS and animation applications by importing data in one format, applying the appropriate translation and exporting them in another format. During the transformation process, as data move from package to package, the researcher gains insight from viewing the different representations of that data.

THE PRODUCTION PROCESS

The production of an animated, 3-D map of Japan census data involved the use of several programs. These were selected based on availability and ability to address the individual components of the cartographic process. The animation process was broken into five stages:

1. Capture and storage of the x-y spatial component, in this case the city, town and village boundaries of Tokyo and four neighboring prefectures. The PC-ArcInfo package was used to digitize and format this data.
2. Addition of thematic variables as the z spatial component and creation of texture maps* for overlay on the resulting 3-D surface. Khoros, on a Sun workstation, provided a visual programming environment conducive to this data transformation process.

---

* In the context of animation, texture map refers to any image which is draped over an object, such as a 3-D surface, giving it color and/or the appearance of texture.

3. Creation and rendering of individual video frames. Wavefront, a powerful animation package on a Silicon Graphics workstation, was used for this stage.
4. Final editing and composition. Macintosh and PC programs were used to create text overlays, backgrounds and other frames for the final editing. Khoros was used to assemble and preview the final video frame images.
5. Transferring final frames to videotape. The final frames were transferred over the campus network to a Diaquest animation controller which buffered and wrote each image to a frame accurate S-VHS recorder.

No single software package was sufficient to perform all these tasks. A combination of tools, with Khoros as the backbone, enabled the transition from paper maps and statistical tables to video tape.

Khoros: Production Backbone

Khoros is an extensible, visual programming system which is useful in a number of ways. It is a powerful data transformation tool as well as data visualizer, capable of creating different views of data and bridging between other, non-compatible software applications. The visual programming environment is flow-based, comprised of a workspace which is populated by input module icons linked to data processing module icons linked to output module icons. Figure 1 shows a workspace for viewing the population density

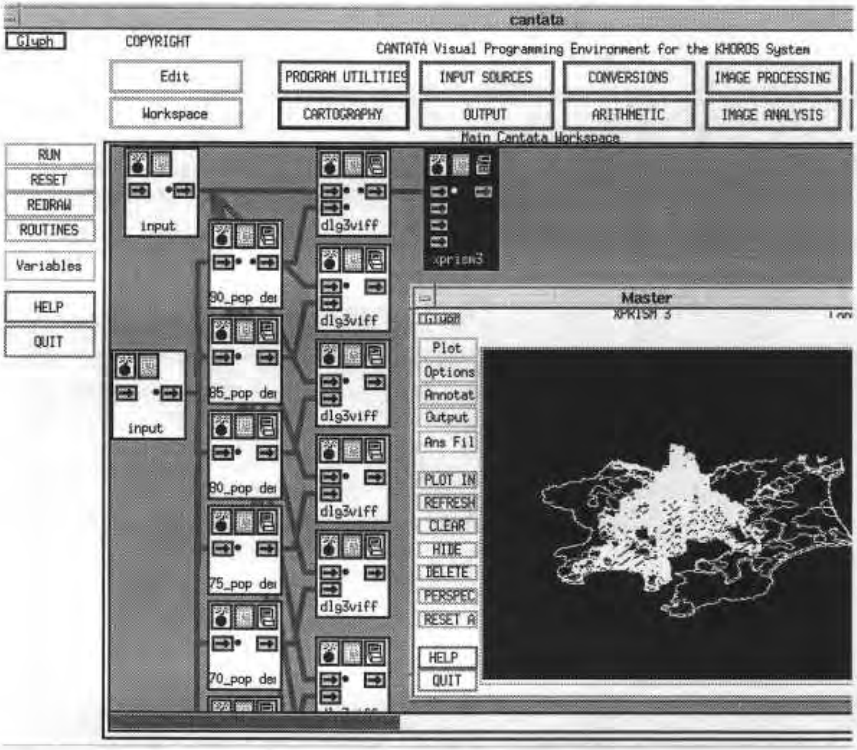


Figure 1. Khoros visual programming environment.

of cities, towns and villages for several census years. Tabular census data are merged with digitized boundaries and fed to a 3-D viewer which is shown displaying the data for 1990 as 3-D contours.

Extending the functionality of Khoros beyond its supplied modules requires C-language programming, but this is greatly assisted by interactive code generators. Two new modules were programmed for this project to input data from PC-ArcInfo and output to Wavefront object files. All other data transformation functions were provided by Khoros in its visual programming environment.

Stage 1: Digitizing

The city, town and village boundary maps for each prefecture were taken from a government data book (Jichisho Gyoseikyoku Shinkoka 1991). These were not ideal for digitizing and assembly since each was drawn at a different scale and did not precisely fit with adjacent prefectures. In addition, they did not contain any coordinate or projection information. This necessitated the use of a GIS, in this case PC-ArcInfo, which could rescale and fit these maps to a single geographic coordinate system. This was accomplished by first digitizing a base map of Japan prefectures* and transforming the digitizer coordinates to projected coordinates.** Then each city, town and village map was digitized and its coordinates transformed to fit the base map. The transformed data were then merged, prefecture boundaries were matched and polygon topology was built into a single coverage for the five prefectures centered around Tokyo city.

The boundary data were exported as a DLG3-Optional (U.S. Geological Survey 1990) formatted file. This format was used because it is supported by Khoros and its major and minor attribute keys fit perfectly with the prefecture and city-town-village ID's.

Stage 2: Assembling Key Data Objects

Census data for each city, town and village area, for the years 1955 through 1990, were entered into a spreadsheet on the Sun workstation. Boundary changes were handled by aggregation or by averaging to normalize the data to the 1991 boundaries. These calculations were handled by the spreadsheet software which also output the ASCII lookup tables which fed into the first Khoros module.

Khoros provides a module to read DLG3 files, but it is deficient in several respects, so a new module was programmed. This new module reads a DLG3-O file and optionally calculates the value of a key from each object's major and minor attribute codes, then uses this key to find a feature value in the lookup table. The data are output as a raster grid with either key values or attribute values written into each cell.

---

* Using a 1:200,000 scale map from the 1990 National Atlas of Japan published by the Japan Geographical Survey Institute.

**PC-ArcInfo does not support the map's oblique-conic projection, so the oblique-mercator projection was substituted. The introduced error was not considered significant for this project at this scale.

Several raster grids were created to produce both x,y,z surface objects and texture maps for input into the Wavefront system. One surface object was created per census year and, due to limitations in the Wavefront system, was limited to a 1 km cell resolution. A second Khoros module was programmed to take the grid and convert it into a rectangular mesh in a Wavefront object format file. Texture maps were created for each census year to color-code areas by population density. Khoros was then used to generate one texture map per video frame by morphing (interpolating) each census year's texture map into the next. These data files were exported to the Wavefront system which rendered the raw animation frames.

Figure 2 shows the surface mesh for 1990 and its associated texture map which has been converted to greyscale for this illustration.

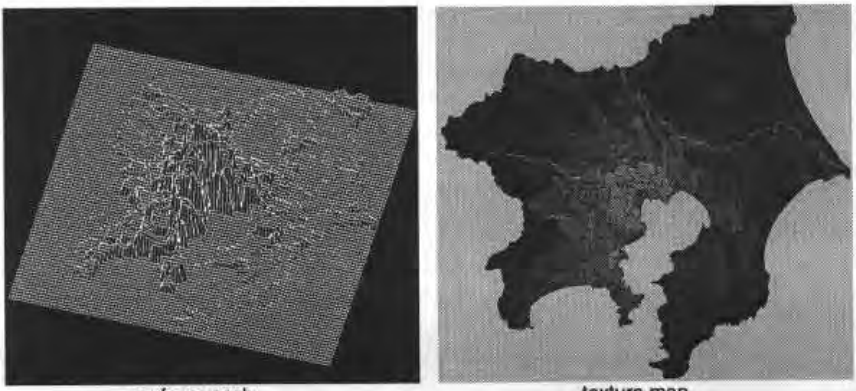


Figure 2. 1990 population density as a surface mesh and texture map

Stage 3: Rendering of Animation Frames

The Wavefront system consists of several modules accessed by a common user interface. The model module was used to link a dummy texture map file to each surface object. Then the preview module was used to create and interpolate between animation key frames. This allowed a combination of object morphing and camera movement. Surface meshes were morphed over time from the 1955 through the 1990 population density surfaces. The camera was kept still during the forward surface morph and moved to another viewpoint while "rewinding" the data.

While in the preview module, it was possible to visualize a rough representation of the final animation. This was useful in determining the best orientation of the camera relative to the 3-D object. Several camera angles, over the same 35 year period, were picked which seemed to show the best views of urban growth from the city center. The first shows southward growth toward Yokohama, the second shows westward growth along a major transportation corridor and the third shows more recent growth in the northeasterly direction.

The creation of the video frames was accomplished by submitting batch process jobs to run over a period of several days. The batch job script substituted the correct Khoros generated texture map for the dummy file before rendering each frame. The output consisted of one raster image file per video frame (30 frames per second of viewing time). These raw, unenhanced frames were written directly to a frame accurate video recorder as a preview step to determine what changes should be made to the animation sequence.

For efficiency, the entire animation was broken into scenes which formed separate sequences of frames. As each scene was completed, its raw frames were transferred back to Khoros for final editing.

Stage 4: Editing and Composing

Introductory and ending title slides were composed and rendered using 3D-Studio on a PC and output in the same file format as the Wavefront rendered frames. Adobe Photoshop on the Macintosh was used to assemble a Landsat image of the data area which had been scanned in four segments. This same software was also used to create pieces of annotation text for pasting onto the animation frames.

Standard Khoros modules were used to interactively paste annotation into sample animation frames as illustrated in Figure 3. Once the layout was determined, these modules were run in a batch job to modify all necessary frames (a process which took several hours per scene).

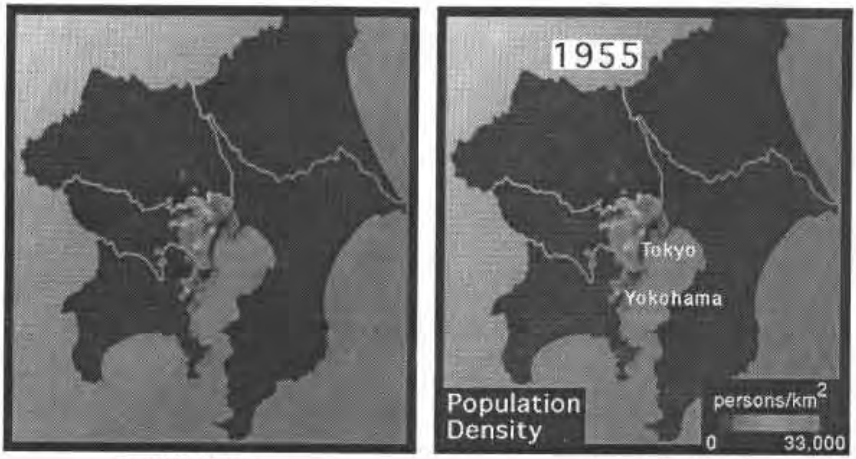


Figure 3. Video frame annotation

Other functions performed by Khoros included dissolving between title slides and dissolving between the introductory Landsat scene and the first data scene. Simple text slides were also assembled by pasting pieces of Photoshop text onto a blank background.

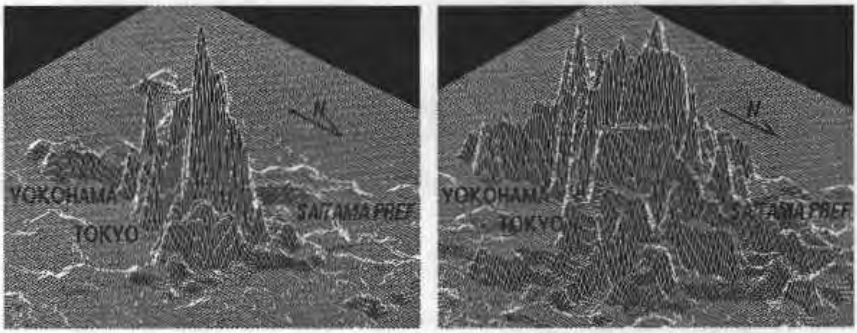
Stage 5: Videotaping

The last stage was the easiest due to the availability of high-end video production equipment. Transferring the animation frame images to S-VHS tape was done with a batch script on the Sun workstation. It sent recording commands and images across the campus network to a Diaquest animation controller which buffered and transferred each image to a frame accurate recorder. Frame accuracy meant that the animation could be transferred in pieces and written to precise locations on tape, a useful feature for making small adjustments. With a transfer rate of just 2 to 3 frames per minute, this step also took many hours per scene.

RESULTS

The final animation, which is less than three minutes long, clearly shows Tokyo city following a classic, doughnut-shaped pattern of growth. The overall pattern reveals an enlarging central core with low population density, surrounded by a widening, high density, suburban ring. However, it is evident that there are sub-patterns within that growth which lead to questions about transportation networks, local area urban planning behavior, land pricing and other factors which influence population growth. For example, the area between Tokyo city and nearby Yokohama city shows relatively higher growth than the area just northeast of Tokyo. Higher growth rate patterns can also be seen moving west and northwest from the city center.

In addition to being able to detect overall trends, animation makes it easier, over static maps, to detect anomalies or outliers which should lead to questions about the area or about the data. For example, in Figure 4, it is difficult to see the growth pattern of cities and towns in the southern region of Saitama Prefecture, north of Tokyo.



1955 Population Density 1990 Population Density

Figure 4. City and town growth north of Tokyo

The animation shows which cities and towns in Saitama appear to be small urban growth areas on their own, as opposed to overflow from Tokyo city. These small urban areas seem to grow as independent peaks, sometimes merging with higher density areas spreading out from Tokyo city.

Adding more dimensionality to the animation will make it possible to visualize the interaction of several variables. For example, using a texture map color-coded by land price or any other variable may reveal relationships between it and the surface variable, population density. Overlaying vector data such as transportation networks and distance indicators would also increase the information content of the display, thereby increasing opportunities for detecting trends and anomalies.

CONCLUSION

Interactive visualization tools are as important to the animation cartographer as they are to the researcher studying the area being mapped. The software packages described in this paper are part of a growing body of enabling technologies which assist us in developing clearer views of the world.

REFERENCES

Choudry, S.I., 1992, A Low Cost Approach to Animated Flow Visualization Using High Quality Video. *Advances in Scientific Visualization*, Berlin, Springer-Verlag, pp. 131-144.

DiBiase, D., A.M. MacEachren, J.B. Krygier, and C. Reeves, 1992. Animation and the Role of Map Design in Scientific Visualization. *Cartography and Geographic Information Systems* 19(4): 201-214, 265-266.

Gershon, N.D., 1992. Visualization of Fuzzy Data Using Generalized Animation. *Proceedings of Visualization '92*, pp. 268-273.

Jichisho Gyoseikyoku Shinkoka, 1991. *Zenkoku shichoson yoran*, Tokyo, Daiichi Hoki.

MacEachren, A.M., and J. Ganter, 1990. A Pattern Identification Approach to Cartographic Visualization. *Cartographica* 27(2): 64-81.

Rasure, J., and M. Young, 1992. An Open Environment for Image Processing and Software Development. *1992 SPIE/IS&T Symposium on Electronic Imaging, SPIE Proceedings* 1659: 300-310.

Sparr, T.M., R.D. Bergeron, and L.D. Meeker, 1993. A Visualization-Based Model for a Scientific Database System. *Focus on Scientific Visualization*, Berlin, Springer-Verlag, pp. 103-121.

Springmeyer, R.R., M.M. Blattner, and N.L. Max, 1992. A Characterization of the Scientific Data Analysis Process. *Proceedings of Visualization '92*, pp. 235-242.

U. S. Geological Survey, 1990. Digital Line Graphs from 1:24,000-Scale Maps: Data Users Guide 1, Reston, U.S. Geological Survey.

OPTIMAL PREDICTORS FOR THE DATA COMPRESSION OF DIGITAL ELEVATION MODELS USING THE METHOD OF LAGRANGE MULTIPLIERS.

M. Lewis & D.H. Smith
Department of Mathematics and Computing, The University of Glamorgan, Pontypridd, Mid-Glamorgan, CF37 1DL., U.K.

Tel.U.K. 443 480480 ext. 2268., Fax: U.K. 443 480553.,
Email : MLEWIS@uk.ac.glam

ABSTRACT

Square or rectangular grids are extensively used for digital elevation models (DEM's) because of their simplicity, their implicit topology and their minimal search time for applications. However, their inability to adapt to the variability of the terrain results in data redundancy and excessive storage requirements for large models. One approach to mitigating this is the use of data compression methods. One such method, based on probabilities, is that of Huffman encoding which gives error-free data compression. The key idea is the use of a model that predicts the data values. The method of Lagrange Multipliers for minimisation of the root mean square prediction error has been applied to local geometric predictors and compared with the least-squares fitting of quadratic and bilinear surface patches. The measure of goodness was the average entropy derived from the differences between the actual and predicted elevations. The lower the entropy, the better is the prediction method. An optimal 8-point predictor proved better than the fitting of polynomial surfaces and gave about a 4% to 7% improvement on a simple triangular predictor.

INTRODUCTION.

The data redundancy inherent in regular grid digital elevation models (DEMs) can be removed by the use of data compression techniques. One common approach is to step through the data values in some predefined order and to make a prediction of the current value from the previous values. The difference between the predicted integer value and the actual value is added to a string of prediction errors, which is encoded using a variable length coding technique such as Huffman encoding [2]. Error free recovery of the original data can be obtained by a reversal of the method. Kidner & Smith [3] proposed a simple triangular predictor for use before Huffman encoding. In this paper, we consider a number of alternative prediction methods.

A MEASURE OF COMPRESSION PERFORMANCE

In general, an estimate of the maximum amount of compression achievable in an error-free encoding process can be made by dividing the number of bits needed to represent each terrain height in the original source data by a first-order estimate of the entropy of the prediction error data. Since there is in general a large degree of redundancy in the source data, an accurate prediction process causes a reduction in the entropy value due to the probability density function of the prediction errors being highly peaked at zero and having a relatively small variance.The mathematical definition of entropy is:

$$H = -\sum_{1}^{N} p(i) \log_2 p(i) \qquad (1)$$

where H is the entropy and p(i) is the probability of each data value. So by estimating the entropy, one can determine how efficiently information can be encoded.

A METHOD OF MINIMISATION.

The method of Lagrange Multipliers for determining maxima and minima of a function S(x,y,z) subject to a constraint condition $\phi(x,y,z)=0$; consists of the formation of an auxiliary function:-

$$G(x,y,z) \equiv S(x,y,z)+\lambda \phi(x,y,z) \qquad (2)$$

subject to the conditions that $\partial G/\partial x = 0$, $\partial G/\partial y = 0$, $\partial G/\partial z = 0$ and $\partial G/\partial \lambda = 0$, which are necessary conditions for a relative maximum or minimum. The parameter λ, which is independent of x,y and z, is called the Lagrange multiplier.
This method has traditionally been used in geostatistical estimation techniques such as kriging [1].

PREDICTION OF TERRAIN ELEVATION DATA.

Given a square or rectangular grid of points {(i,j): i=0,1,...,N; j=0,1,...,M} we will let Z denote Z(i,j), the point being predicted and take Z_1= Z(i-1,j), Z_2 = Z(i-1,j-1), Z_3= Z(i,j-1). We use the values of Z_1, Z_2, Z_3 in a predictor of the form:

$$\text{pred}(Z) = \text{Nearest Integer } \{ \mu_1 Z_1 + \mu_2 Z_2 + \mu_3 Z_3 \}. \qquad (3)$$

The greatest compression will be achieved if the entropy of the set of values { Z - pred(Z) ; i=1,2,...N; j=1,2,...M } is minimised. Although the form of the expression for entropy makes minimisation difficult, we can attempt the minimisation indirectly as follows:
(1) Assume that the mean error {Z - ($\mu_1 Z_1 + \mu_2 Z_2 + \mu_3 Z_3$) } is zero;
(2) Subject to this constraint, minimise the squares of the errors

$$S(\mu_1,\mu_2,\mu_3) = \Sigma(Z - \mu_1 Z_1 - \mu_2 Z_2 - \mu_3 Z_3)^2 \qquad (4)$$

where the summation is over all terrain height values i=1,2,...,N; j=1,2,...,M.

Then $S(\mu_1,\mu_2,\mu_3)$ becomes:

$$\Sigma_{ij} Z^2 + \Sigma_{ij} \mu_1^2 Z_1^2 + \Sigma_{ij} \mu_2^2 Z_2^2 + \Sigma_{ij} \mu_3^2 Z_3^2 - 2\Sigma_{ij} Z Z_1 \mu_1 - 2\Sigma_{ij} Z Z_2 \mu_2 -$$
$$2\Sigma_{ij} Z Z_3 \mu_3 + 2\Sigma_{ij} Z_1 Z_2 \mu_1 \mu_2 + 2\Sigma_{ij} Z_1 Z_3 \mu_1 \mu_3 + 2\Sigma_{ij} Z_2 Z_3 \mu_2 \mu_3 .$$

In order that $S(\mu_1,\mu_2,\mu_3)$ be minimised subject to the mean error being zero, we let

$$G(\mu_1,\mu_2,\mu_3) = S(\mu_1,\mu_2,\mu_3) + \lambda(\Sigma_{ij} Z - \mu_1 \Sigma_{ij} Z_1 - \mu_2 \Sigma_{ij} Z_2 - \mu_3 \Sigma_{ij} Z_3) \qquad (5)$$

and set the partial derivatives
$\partial G/\partial \mu_1$, $\partial G/\partial \mu_2$, $\partial G/\partial \mu_3$, $\partial G/\partial \lambda$ to zero; i.e.:

$$2\mu_1 \Sigma_{ij} Z_1^2 - 2\Sigma_{ij} Z Z_1 + 2\Sigma_{ij} Z_1 Z_2 \mu_2 + 2\Sigma_{ij} Z_1 Z_3 \mu_3 - \lambda \Sigma_{ij} Z_1 = 0.$$

$$2\mu_2 \Sigma_{ij} Z_2^2 - 2\Sigma_{ij} Z Z_2 + 2\Sigma_{ij} Z_1 Z_2 \mu_1 + 2\Sigma_{ij} Z_2 Z_3 \mu_3 - \lambda \Sigma_{ij} Z_2 = 0.$$

$$2\mu_3 \Sigma_{ij} Z_3^2 - 2\Sigma_{ij} Z Z_3 + 2\Sigma_{ij} Z_1 Z_3 \mu_1 + 2\Sigma_{ij} Z_2 Z_3 \mu_2 - \lambda \Sigma_{ij} Z_3 = 0.$$

$$\Sigma_{ij} Z - \mu_1 \Sigma_{ij} Z_1 - \mu_2 \Sigma_{ij} Z_2 - \mu_3 \Sigma_{ij} Z_3 = 0.$$

Define the coefficients C as:

$C_{11} = \Sigma_{ij} Z(i-1,j)^2$

$C_{12} = \Sigma_{ij} Z(i-1,j) \times Z(i-1,j-1)$

$C_{13} = \Sigma_{ij} Z(i-1,j) \times Z(i,j-1)$

$C_{22} = \Sigma_{ij} Z(i-1,j-1)^2$

$C_{23} = \Sigma_{ij} Z(i-1,j-1) \times Z(i,j-1)$

$C_{33} = \Sigma_{ij} Z(i,j-1)^2$

$C_{01} = \Sigma_{ij} Z(i,j) \times Z(i-1,j)$

$C_{02} = \Sigma_{ij} Z(i,j) \times Z(i-1,j-1)$

$C_{03} = \Sigma_{ij} Z(i,j) \times Z(i,j-1)$

$C_0 = \Sigma_{ij} Z(i,j)$

$C_1 = \Sigma_{ij} Z(i-1,j)$

$C_2 = \Sigma_{ij} Z(i-1,j-1)$

$C_3 = \Sigma_{ij} Z(i,j-1)$.

The equations reduce to:-

$$\mu_1 C_{11} + \mu_2 C_{12} + \mu_3 C_{13} + (-\lambda/2) C_1 = C_{01}$$

$$\mu_1 C_{12} + \mu_2 C_{22} + \mu_3 C_{23} + (-\lambda/2) C_2 = C_{02}$$

$$\mu_1 C_{13} + \mu_2 C_{23} + \mu_3 C_{33} + (-\lambda/2) C_3 = C_{03}$$

$$\mu_1 C_1 + \mu_2 C_2 + \mu_3 C_3 + (-\lambda/2).0 = C_0.$$

Once all the coefficients C have been calculated, the problem reduces to solving 4 linear equations in 4 unknowns. The solution vector $[\mu 1, \mu 2, \mu 3, -\lambda/2]$ gives the fitting coefficients. The term involving the Lagrange multiplier (λ) is not required in the prediction process.

The above equations have been set up for predicting a value for a point Z(i,j) based on 3 nearby terrain heights. The same method can be used for predicting a value for the same terrain height from 8 neighbouring heights with solution vector $[\mu_1,\ldots,\mu_8,-\lambda/2]$ giving the fitting coefficients to Z(i-1,j), Z(i-1,j-1), Z(i,j-1), Z(i-2,j), Z(i-2,j-1), Z(i-2,j-2), Z(i-1,j-2) and Z(i,j-2), which are shown as points 1...8 in Fig. 1.

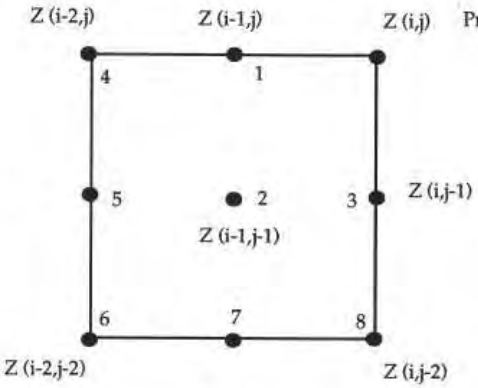


Fig. 1.

APPLICATION TO DIGITAL ELEVATION MODEL DATA.

In this section we will compare the entropy values for our 3-point and 8-point predictors with the triangular predictor of Kidner and Smith [3] which is given by pred(Z)= Z(i-1,j) - Z(i-1,j-1) + Z(i,j-1). Given a square or rectangular grid the first row and column (or first two rows and columns for an 8-point predictor) are excluded and the points are scanned column by column starting from Z(1,1) (or Z(2,2) for an 8-point predictor). For each point the prediction is calculated and the error pred(Z)-Z is recorded. From the frequencies of these errors, the probabilities of the errors and hence the entropy can be calculated.

We will use two British Ordnance Survey 401x401 Digital Elevation Model grids consisting of points sampled at 50 metre intervals accurate to the nearest metre. Source data is held on disk as 2-byte integers (16 bits). ST06 is an area of South Wales containing sea and land areas to the south and west of Cardiff and ST08 covering the Taff and Rhondda Valleys centred near the town of Pontypridd. The terrain profiles consist of both smooth and sharp changes in topology, i.e. deep valleys and rounded hills in ST08 and areas with smoother gradients but containing coastal cliffs in ST06. The original data was rounded to units of 2 metres as the elevation range then allowed all elevations to be represented in 8 bits for convenient comparison.

The entropies, values of S^2 and μ-values are given in Table 1. For the triangular predictor, the 3-point predictor and the 8-point predictor, the entropies are 1.3910, 1.3581, 1.3042 bits per elevation for ST06 and 2.3689, 2.3465 and 2.2577 bits per elevation for ST08. These values should be compared with the 8 bits per elevation of the original data.

Optimum Predictors for a Digital Elevation Model.

Predictor	Data Set	(S^2)	Coefficients (μ -values)	Entropy (bits/elevation)
Triangular [3]	ST06	9.7596×10^4	1,-1, 1	1.3910
3pt	ST06	8.0336×10^4	0.8267 - 0.6967 0.8703	1.3581
8pt	ST06	6.9809×10^4	0.8237 - 0.4168 0.9829 - 0.1169 - 0.0866 0.0824 - 0.0773 - 0.1912	1.3042
Triangular [3]	ST08	3.10961×10^5	1,-1, 1	2.3689
3pt	ST08	2.87193×10^5	0.9380 -0.8610 0.9230	2.3465
8pt	ST08	2.46358×10^5	1.0370 -0.5736 0.9236 -0.2214 -0.0313 0.0585 -0.0263 -0.1666	2.2577

Table 1.

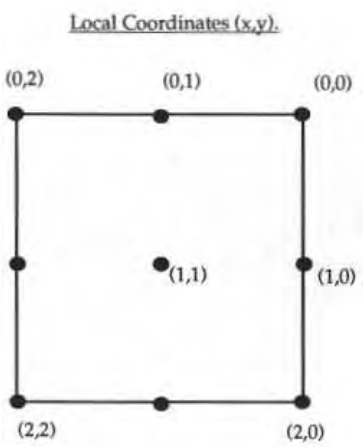


Fig. 2.

For each surface patch used to fit to a set of six points, we can work within a local coordinate system as in Fig. 2. This has the advantage that the matrix only has to be inverted once. The point to be predicted will simply have the coordinates (0,0) and so (0,0) is substituted into F(x,y). Then the difference between the value of the predicted height using this method and the actual height value is calculated and rounded to the nearest integer. This procedure is repeated over the whole terrain data matrix and stored as a string of difference values.
As for the Lagrange multiplier method, the average entropy of the resulting string of corrections is calculated. The results are presented in Table 2.

PREDICTION BY FITTING QUADRATICS THROUGH SETS OF TERRAIN HEIGHTS.

The least squares fit of a quadratic function defined by $a+bx+cx^2$ through three sets of three points in a west-east, south-north and a south-west to north-east direction was done to predict the point Z(i,j). (These are illustrated by ray 1, ray 2 and ray 3 in Fig. 3). This method would enable the capture of surface convexity or concavity. The actual predicted value was taken either as the median value of the three quadratics or as the average predicted value of the three quadratics. The resulting prediction errors in both cases were used to calculate the entropy as in the case of a least-squares surface fit.
A similar procedure was followed of retaining the first row and column of terrain height values together with the second and third row and column as prediction errors using the triangular predictor [3]. The errors arising from both the mean and the median prediction for the three quadratics are stored separately. In each case the entropy for these errors is calculated.
In this case we can either substitute in points to calculate the coefficients directly or minimise the least squares error as before. We do this for each ray separately.

$$L_n=\Sigma_{i=1..3}\ (Z_i - a-bx-cx^2)^2 \text{ where } Z_i = \text{height } (x_i). \qquad (9)$$

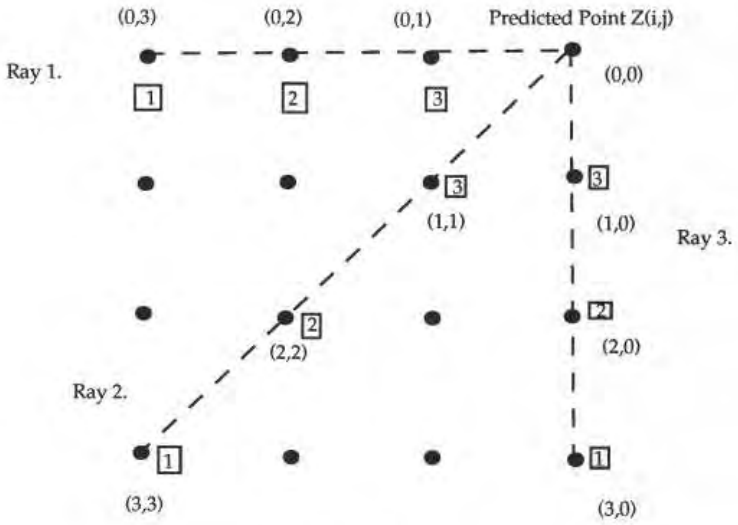


Fig. 3.

In matrix form this becomes:

$$\begin{bmatrix} 3 & \sum_i x_i & \sum_i x_i^2 \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 \\ \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_i Z_i \\ \sum_i x_i Z_i \\ \sum_i x_i^2 Z_i \end{bmatrix}$$

The solution vectors [a,b,c] give the fitting coefficients for each quadratic equation used to determine the predicted height. In this case, we have used three rays with each ray consisting of three contributing terrain height values.

The results for our representative test data were as follows: In ST06, the entropy taking the mean value of the ray predictions was 2.1692 bits and taking the median value 2.5885 bits. In ST08, the respective entropy values were 3.9579 and 4.2243 bits per elevation.

PREDICTION BY BILINEAR SURFACE FITTING.

In this method, a bilinear surface defined by the equation $a+bx+cy+dxy$ is used to fit through four points P,Q,R,S and used to predict the five points A,B,C,D,E and the differences between the actual and predicted values stored in a matrix (see Fig.4). The procedure is repeated on the square P,C,E,A using the true height

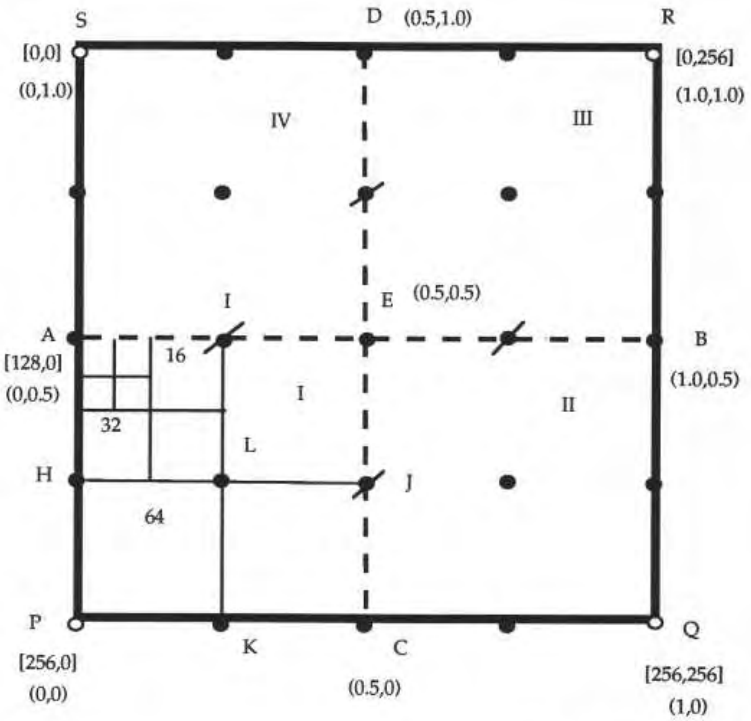
values to predict the points H,I,J,K,L in the diagram. When a predicted point is on the boundary of two separate surface patches, the mean value of the combined prediction error or correction value is taken. The initial array of terrain heights is a 256x256 array (257x257 height values) and is recursively sub-divided into quadrants of 128x128, 16 of 64x64 and 64 of 32x32 and so on. During this recursive subdivision, the above prediction process is applied to each quadrant where on each subdivision the four mid-points and centre point are predicted by fitting an increasingly finely grained bilinear surface. Once again, each surface patch is defined by a local coordinate system i.e. if P,Q,R,S have coordinates (0,0), (1,0), (1,1), (0,1) respectively then each predicted point in each quadrant at each sub-division will have the local x,y coordinates as illustrated by A,B,C,D,E of (0,0.5), (1,0.5), (0.5,1), (0.5,1) and (0.5,0.5).The four coefficients of each bilinear surface used to define each surface patch are calculated by a system of 4 linear equations formulated by the least squares minimisation procedure described above or can be done by direct substitution where each predicted point is:

$$P+(Q-P)x+(S-P)y+(P-Q-S+R)xy. \qquad (10)$$

The algorithm terminates when the recursion has produced 4^7 smaller patches of 3x3 elevation points. Since our test-data, ST06 and ST08, are fixed sized arrays of 401x401 points, the recursive sub-division algorithm is run on 4 overlapping tiles of 257x257 points with origins at coordinates (0,0), (0,144), (144,0), (144,144) for the maximum terrrain data matrix coverage. In ST06 the entropy values were 2.0919, 1.8117, 2.9287, 2.7503 bits for the overlapping segments. The corresponding values for ST08 were 3.6112, 3.8811, 3.5570 and 3.6358 bits respectively.

DISCUSSION.

The results above show quite clearly that a small improvement can be made to the simple triangular predictor method for both the three-point and eight-point predictors by the minimisation method of Lagrange multipliers. Typical savings in the average entropy values varied between 4% and 7%. It was interesting to note that in both the 3-point and 8-point prediction methods, the significant coefficient (μ) values affecting the prediction of the terrain height always corresponded to points 1,2 and 3 with a smaller contribution from points 4,6 and 8 for the 8-point predictor. This seemed to support the rationale behind the triangular predictor [3]. Comparisons with a least squares fit of six points to predict the same point and different least squares fits of both a quadratic interpolation of neighbouring points and a bilinear surface interpolation have confirmed this. These surface fitting prediction methods failed to achieve a lower entropy value with the least squares fit being the best of the other three methods. Table 2 shows comparative entropy values for our data sets ST06 and ST08 for the different prediction methods described. For many data sets compression ratios above 4 or 5 are easily achievable using a error-free Huffman encoding algorithm with minor modification to the code given in [3] to include the calculation of the coefficients (μ-values) for either the 3-point or 8-point predictor.



[0,128] - Array Coordinates
(0.5,0)-Local Coordinates

I..IV Quadrants

Terrain Height
Predicted Height
Boundary Point

Points A,B,C,D,E are predicted from P,Q,R,S. by fitting a bilinear surface a+bx+cy+dxy. The points H,I,J,K,L are predicted from surface fit to A,E,C,P. The process is repeated in quadrants II, III, IV to determine mid-points and centres of quadrants. Points I and J are predicted from 2 separate bilinear surface fits i.e point I from quadrants I and IV and point J from quadrants I and II. In such cases, the average correction (prediction error) is taken from the separate prediction estimates.

The smallest quadrant is a 3 x3. The total no. of squares is 16,384 ($=4^7$).

Figure 4.

COMPARISON BETWEEN PREDICTION METHODS		
Prediction Method	Entropy (bits/elevation) ST06	ST08
Triangular Predictor [3]	1.3910	2.3689
3-point	1.3581	2.3465
8-point	1.3042	2.2577
Least Squares Surface Fit	2.0521	3.4159
Quadratic Fit (mean result)	2.1692	3.9579
Quadratic Fit (median result)	2.5885	4.2243
Bilinear Surface Fit (256x256)		
Origin at: (0,0)	2.0919	3.6112
(0,144)	1.8117	3.8811
(144, 0)	2.9287	3.5570
(144, 144)	2.7503	3.6358

Table 2.

CONCLUSIONS.

The 3-point and 8-point predictors are clearly superior to the other methods reported. With all predictors, the Huffman code gives an average code length slightly greater than the entropy. It is also necessary to store a code table, a look up table for efficient decoding, the coefficients for the optimal predictors and the first row and column(or first two rows and columns for the 8-point predictor) [3]. However for large models, the additional storage requirements are very small.
An alternative approach to terrain compression of a regular grid data is a transform technique - the two-dimensional discrete cosine transform (2D-DCT). Transform coding allows greater compression but is computationally intensive and gives some error on reconstruction of the data. Transform coding can be combined with Huffman encoding or Run Length Encoding to allow further compression. As an example, terrain data has been compressed by a factor of 18.75:1 with this method by adapting published algorithms [4]. In this case, the reconstructed data is about 75% error-free.

ACKNOWLEDGEMENTS.

This work has been undertaken with sponsorship by the Science and Engineering Research Council (SERC) in collaboration with the Defence Research Agency, Malvern, U.K.

REFERENCES.

[1] Dowd P.A. "A Review of Geostatistical Techniques for Contouring" in Fundamental Algorithms for Computer Graphics. Ed E.A. Earnshaw . NATO ASI Series Vol. F17, 1985, Springer-Verlag.
[2] Huffman D.A. "A Method for the Construction Of Minimum Redundancy Codes". Proc. IRE, Vol. 40, 1952. pp. 1098-1101.
[3] Kidner D.B. & Smith D.H. "Compression of Digital Elevation Models by Huffman Encoding"; Computers & Geosciences, Vol. 18, No. 8, pp. 1013-1034, 1992.
[4] Nelson M.R. The Data Compression Book. Prentice Hall. 1991.

ON THE INTEGRATION OF DIGITAL TERRAIN AND SURFACE MODELING INTO GEOGRAPHIC INFORMATION SYSTEMS

Robert Weibel
Department of Geography
University of Zurich
Winterthurerstrasse 190
CH-8057 Zurich (Switzerland)
weibel@gis.geogr.unizh.ch

ABSTRACT

Current GIS are still predominantly oriented towards the processing of planimetric (2-D) data. On the other hand, there is an increasing need for capabilities to handle 2.5-D and 3-D data in many GIS applications. Based on the example of digital terrain modeling, this paper discusses the requirements and possibilities for the functional and database integration of digital terrain models (DTMs) into GIS. Functional integration based on a unified user interface and file structures today is implemented in several GIS. Current solutions for database integration, however, are prohibitively inefficient and the useful properties of DBMS cannot yet be exploited given the current state of research in spatial databases. Future research should foremost focus on the development of more efficient database schemata for DTM data structures. Special consideration should be given to data structures that can be used for disk-based triangulation algorithms, as they can provide an equivalent representation in memory and secondary storage. A further problem is the transfer of DTMs between different systems. While the transfer of geometry and topological relations could be implemented relatively easily through an extension of current transfer standards, the transfer of special local adaptations and interpolation methods that model the surface is non-trivial. Methods of object-oriented databases that allow the definition of persistent objects could offer a solution.

INTRODUCTION

In recent years, an increasing need has been expressed in the literature for an extension of planimetric (2-D) geographic information systems (GIS) to accommodate 2.5-D and even 3-D modeling functions. Based on the example of digital terrain and surface modeling, this paper looks at the requirements and possibilities for integrating such functions into planimetric GIS at the functional and at the database level. Integration involves two aspects: (1) *functional integration*, the transparent use of 2-D and 2.5-D or 3-D functions under common data formats and a unified user interface; and (2) *database integration*, the permanent storage and management of DTM and planimetric data under a common data management scheme, possibly a common database.

A digital terrain model (DTM) provides a 2.5-dimensional digital representation of a portion of the Earth's surface. By 2.5-D it is commonly meant that the topographic surface is represented as a field, having unique z-values over x and y, rather than providing a true 3-D surface or volume representation. The concept of DTMs is not restricted to topographic surfaces, but can be used to represent other continuously varying phenomena such as stratigraphic bedding planes, air temperature, or population density. Because of their great importance in many GIS applications, DTMs are taken as an example in this paper to discuss the integration of non-planimetric functionality into planimetric GIS. However, many of the observations made here also apply to 3-D modeling, if at a higher level of complexity.

The problem of permanent storage of DTMs and their integration into GIS databases is rarely treated in the literature. Emphasis is still on memory-based algorithms and data structures rather than on the permanent storage of DTMs. However, given the growing size of current terrain modeling projects and the general trend towards the exploitation of database technology in GIS, problems of storage and management of terrain-related data gain increased importance.

The predominant approach for DTM integration is still based on file structures relating to the data structures used in memory, and coupling of terrain modeling functions with the other GIS functions through a common user interface. Thus, while DTMs are functionally rather well integrated into some GIS, database integration is most often lacking completely. More recently, some authors have advocated the full integration of DTMs into the database system of the GIS, which would potentially allow enhanced retrieval and querying functionality. While some of these papers (e.g., Fritsch and Pfannenstein 1992, Molenaar 1990, Höhle 1992) helped to cover some of the theoretical basis of DTM integration, they still fall short of providing practicable solutions. Various issues have to be considered in order to develop efficient schemes for data integration of DTMs into GIS that actually meet the functional requirements of different DTM applications.

Four main topics will be addressed here: Which data representations exist in a DTM system? What are the requirements of different applications with respect to DTM integration? What are the alternatives for DTM storage and integration? How can terrain-related data be transferred between different systems?

MODELING COMPONENTS OF A DTM SYSTEM

Functional Tasks of Digital Terrain Modelling

The following five general tasks can be distinguished within a digital terrain modeling system:

- DTM generation: Sampling of original terrain observations (data capture), and formation of relations among the diverse observations (model construction) to build a DTM.
- DTM manipulation: Modification and refinement of DTMs, and the derivation of intermediate models.
- DTM interpretation: Analysis of DTMs, information extraction from DTMs.
- DTM visualization: Display of elements and properties of DTMs and derived information.
- DTM application: Development of appropriate application models for specific domains (e.g., models for erosion potential, surface water runoff, or noise pollution) to make efficient and effective use of terrain modeling techniques.

These five tasks should be treated as interrelated components of the terrain modeling process; they form elements of a chain that can only be as strong as its weakest link. Thus, a comprehensive DTM system should attempt to support all tasks of terrain modelling equally well, and be adaptive to given problems.

The above scope of a DTM system defines the range within which functional requirements of individual application domains can be specified.

Data Representations within a DTM System

Usually, DTMs are associated with data structures such as regular grid, triangulated irregular networks (TINs), contours, or surface patches. However, besides these representations, other terrain-related models exist in a DTM system, each of them necessitating particular data structures and specific treatment.

The input for the generation of DTMs is given by original observations that are sampled as point, line, or polygon features (e.g., spot heights, breaklines, con-

tours or dead areas). Since they form the foundation of the DTM, they can be termed the *basic model* (Weibel and Heller 1990). The management of these features in the GIS database is simple, as long as the database schema allows to handle z-coordinates for points, lines, and polygon outlines. The basic model reflects the best state of knowledge obtainable for a given project and is thus stored permanently. Any other representation in a terrain modeling project can be derived from this basis, if the rules for generating derived models are known.

Various analytical and visual products can be derived from DTMs through interpretation and visualization (e.g., drainage basins, viewsheds, contours, perspective views, etc.). The models that can be used to represent these products normally are equivalent to those used for corresponding 2-D data. Thus, functional and database integration of products derived from DTMs is easily possible. Again, however, it must be possible to represent z-coordinates in the database (e.g., to store the 3-D coordinates of drainage channels or basins).

DTMs turn the basic model into a description that is useful for surface modeling. Of the various data structures are possible for DTMs, the two classes of regular grid, and triangulated irregular network (TIN) data structures are predominant. Since most classes of data structures used for DTMs do not relate to those commonly employed for handling planimetric features, different solutions must be found for the storage and management of these models. The question then is how to integrate DTMs with other representations, specifically with the basic model. In other words, the problem is how to maintain consistency of representations across different models. Since different DTM applications have different requirements, it can be expected that alternative solutions are necessary for DTM integration.

DTM INTEGRATION: REQUIREMENTS

The general requirements of a DTM system that is used in conjunction with a GIS typically include a wide range of criteria (the terms used below are partly based on Meyer 1988):

- Functionality: the ability to support a wide range of applications. This is probably the single most important criterion for users interested in getting their job done.
- Efficiency: the good use of the available resources, both in terms of storage space and computing time. This criterion is especially important when large amounts of data need to be processed or fast response times are required for high interactivity. DTMs today easily involve several 100,000 data elements.
- Correctness: the ability to deliver a correct solution for a given problem (or, alternatively, inform the user about the inability to do so).
- Consistency: the ability to avoid or resolve conflicts between multiple representations of an entity. A conflict may, for instance, arise when an element in the basic model is removed, but this change is not propagated to the DTM which had been generated previously from these input data.
- Robustness: the ability to resolve special cases in the input data or at least fail gracefully. This requires sound exception handling strategies.
- Compatibility: the ability to transfer data between different representations or systems, minimizing information loss. This issue mainly relates to the transfer of complex data structures (e.g., TINs) between different systems.
- Adaptivity: the ability to adapt to surface character and modeling purpose. This requirement also necessitates the ability to support different representations of a phenomenon – different data structures or interpolation schemes – and to transform between different representations.
- Extendibility: the ease with which the data may be adapted to changes in specifications. Updates of the data may occur over time, necessitating version

management strategies and procedures to integrate different states into a single coherent and consistent model.

- Security: the ability to control access to the data selectively for individual users. This aspect is not essential for all applications. It may be sufficient to rely on the security mechanisms of the operating system.
- Concurrency: the ability to manage data access of multiple users at the same time. This criterion can be important for large installations with many concurrent users.
- Ease of use: the ability to let users interact with the software in an intuitive and flexible way. Significantly contributes to the performance users can get from the system in terms of functionality, efficiency, and correctness.
- Ease of integration: the ability to integrate DTMs as well as products derived thereof with other data in a GIS. This criterion has a major impact on the functionality and ease of use of a GIS package, and also influences the efficiency of workflows involving multiple steps.

Of course, the actual requirements which are expected from a specific system depend on the particular target applications and are usually only a subset of the above list. Some criteria are essential for all applications: ease of use and efficiency are always desirable, correctness and robustness are always required, and adaptivity to varying data characteristics is fundamental to ensure correctness. Consistency is important in the context of DTMs, because they always involve multiple representations (basic model, grids, TINs, etc.).

Some applications have very specific requirements. For instance, a DTM system for use in opencast mining requires efficiency in order to keep track of changes of the terrain surface, and necessitates extendibility and consistency of multiple representations of the surface at different points in time. An institution which is involved in data production and distribution assigns high priority to the compatibility of DTM data formats. Security and concurrency are commonly restricted to environments where controlled multi-user access is required (e.g., large institutions with personnel from different departments working concurrently on the same data).

With respect to the criteria relating to the use of DBMS for managing DTMs – consistency, extendibility, security, and concurrency – two broad classes of application domains can be distinguished. The first class consists of applications that focus on DTM generation and manipulation, tasks that are usually important to institutions collecting, editing, managing, and distributing terrain-related data. The second group of applications concentrates on information extraction, that is, on DTM interpretation and visualization. Examples of such uses are the computation of gradient and aspect information (e.g., for the estimation of the potential for soil erosion) or the production of contour maps or block diagrams. These applications typically operate as batch-like processes, and do not require access to individual elements of a DTM.

While the first class of applications could directly benefit from a database integration of DTMs into GIS, requirements of the second class with respect to DTM storage and management are rather limited. For instance, it is very rare that a user would want to know the gradient of a particular triangle of a TIN.

DTM INTEGRATION: POSSIBLE STRATEGIES

Maintenance of Consistency

As noted above, the maintenance of the consistency between the basic model and derived DTMs is a major concern, independently of the data structures (grid, TIN) that are used to represent the DTMs. Inconsistencies may be introduced whenever modifications to the shape of the terrain surface become necessary.

Possible cases include the addition of new terrain observations to improve the surface model, or the removal of erroneous data elements.

If both the basic model and the derived DTMs remain static and no changes occur, there is no danger that the consistency between the different representations is violated, even if they are stored independently from each other. However, if modifications to the contents of one or the other of the models are made, the surface will be represented differently depending on which model is used, and steps have to be undertaken to prevent or eliminate these discrepancies. Usually, the modification of a single feature in the basic model will induce changes to more than one element of the associated DTMs, since surface gradients in the vicinity of the affected feature will also change.

A practical approach to maintaining consistency is to regard DTMs (grids, TINs) as derivative products from the basic model, and to focus all editing operations on the elements of the basic model. The DTMs are used as a visual backdrop in the editing process (e.g., via contours, hillshading, or stereo) to give an impression of the form of the continuous surface, which the basic model obviously cannot. This approach greatly simplifies the maintenance of consistency between models. Change propagation occurs from one model to many, instead of from many to many models. Changes can then be propagated from the modified basic model to the associated DTMs through either global regeneration (building the entire models anew) or, preferably, local modifications involving only those elements which are affected. Functions for editing of DTMs are discussed in Weibel and Heller (1991).

Alternatives for DTM Integration

Loose coupling: The simplest form of DTM integration consists of a loose coupling mechanism between the DTM system and the GIS via common data interfaces. While the two systems operate independently, this mechanism allows to export data sampled in a GIS to the DTM system to generate a surface model, and import back the results of surface analyses. This approach is frequently found in GIS that do not dispose of a built-in terrain modeling module, but offer an interface to DTM software systems provided by other vendors instead. The most serious drawbacks of this strategy are that queries across systems are difficult and inefficient, communication is usually restricted to the exchange of entire data sets, and inconsistencies can easily be introduced between representations in the two systems.

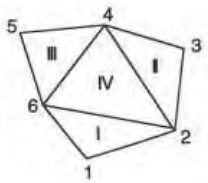
Functional Integration: The next step is to incorporate the software module for terrain modeling into the GIS under a unified user interface and common data formats, offering better opportunities for functional integration.

This approach first of all has implications with respect to the user interface. While most batch-like functions can be handled by the same user interface designs as for 2-D GIS, interactive operations require specific control mechanisms. Examples include special manipulation controls for the interactive manipulation of projection parameters (viewpoint, view direction, camera parameters, etc.) of perspective displays and flight path design for animations (Hussey et al. 1986), or interaction tools for interactive surface and volume modeling and sculpting (Galyean and Hughes 1991). Depending on the complexity of the modeling operations that are involved, these 3-D control mechanisms are not just simple extensions of their 2-D equivalents.

In terms of storage schemes that are used to functionally integrate DTM systems into GIS, file-based structures are still the predominant method used today. The simplest file-based approach stores the entire information of a DTM in a single file. Usually, however, this approach is only practical for simple gridded DTMs. More complex representations such as TINs are predominantly handled in multi-

file structures, with multiple files relating to internal data structures such as the winged triangle structure (Fig. 1) which is used by various commercial systems. Usually, some header information is added to the files or an additional header file is created, including descriptive information which summarizes the characteristics of the DTM such as the number of vertices, number of triangles, number of hull vertices, number of breaklines, spatial extent, and data source. Figure 1 shows an example of a schema comprising three files that would typically result using the winged triangle data structure. Optionally, a header file with descriptive information and a file containing the vertices of the triangulation hull can be included. If programming languages are used that allow positioning in the byte stream of a file (e.g., Fortran, C, or C++), then the storage, retrieval and update of individual vertices or triangles is possible. While the winged triangle representation is widely used in TIN-based systems, it should be noted that other data structures are possible which are more compact or more efficient, or lend themselves better to disk-based algorithms (Heller 1990).

In case the carrier GIS makes use of a DBMS to store spatially referenced data, file-based storage schemes can easily be extended by storing descriptive information about DTMs in the database. That is, the actual DTMs are stored in single or multiple files, while the files names (including the path to their location in the file system) and the descriptive statistics are held in the database. This descriptive information can easily be stored as attributes in a DBMS. Via the query language of the GIS, DTMs can be retrieved, and queries can be answered such as 'Which DTMs of resolution better than 30 m are available within the city limits of Zurich?'. However, access through the DBMS is still restricted to DTMs as a whole. If individual elements of terrain models are to be retrieved, this must be accomplished by specific functions embedded in the DTM module. Also, advantageous features of DBMS such as security and recovery mechanisms cannot be exploited.



Triangle adjacencies	
tri_nr	neighbors
I	0, IV, 0
II	0, 0, IV
III	0, 0, IV
IV	II, III, I

Vertices	
tri_nr	vert_nr
I	1, 2, 6
II	2, 3, 4
III	4, 5, 6
IV	2, 4, 6

Vertex coordinates			
vert_nr	x	y	z
1	x1	y1	z1
2	x2	y2	z2
3	x3	y3	z3
4	x4	y4	z4
5	x5	y5	z5
6	x6	y6	z6

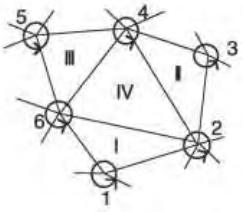
Fig. 1: Winged triangle data structure

Full Database Integration: Over the past decade, the majority of developers of commercial GIS have turned towards the use of commercial DBMS – primarily relational DBMS – to support the storage and retrieval of spatially referenced data (Healey 1991). For simplicity, the discussion here is restricted to relational DBMS (RDBMS), although other types exist (Healey 1991). Modern RDBMS offer a number of useful capabilities such as built-in query languages for retrieval of data elements, facilities for report generation, security mechanisms to limit access for unauthorized users, multi-user support, and rollback recovery mechanisms in case of system failure. On the other hand, the use of database technology for spatial data requires several important extensions to be made to standard RDBMS, relating to query languages, access methods, and database schemata (Frank 1988, Healey 1991, Haas and Cody 1991).

With respect to the integration of DTMs into RDBMS, the single most critical issue is probably the design of the database schema, since the time needed to access individual DTM elements increases with each table that is added and with each relational join that needs to be formed between tables. As a prerequisite to DTM

integration, the database schema must be capable of holding x, y, *and* z-coordinates for the point, line, and polygon features of the basic model. (At this point, it should be noted that several commercial GIS products still do not meet this requirement, in which case 'workarounds' must be provided by storing spot heights, breaklines, and dead areas in text files.) The further discussion is restricted to database schemata for TINs, which are more complex to handle than grids. Possible solutions for grids are presented in Waugh and Healey (1987).

To store a TIN in a GIS database, it would be possible to treat triangles as a special case of polygons, with straight arcs and nodes that have z-coordinates (Waugh and Healey 1987). However, this strategy could only serve as a brute-force approach, as the specific topological properties of triangulations are not adequately modelled. Another possibility is to base the schema on the winged triangle data structure of Figure 1. Instead of the three files that were used for the file-based storage method discussed above, three tables are defined for the triangles, vertices, and vertex coordinates. This schema is, for instance, used in the system described by Steidler et al. (1990). A benefit of this scheme is that information about the individual TIN elements (topological relations and coordinates) can be readily queried. A further advantage is that points of the basic model and vertices of the TIN are made equivalent: they are stored only once in the table representing the codes of the sample points and TIN vertices, respectively. This mechanism guarantees that if a vertex is removed from the TIN, the corresponding point is also removed from the basic model. The reverse, however, is not true. If a point is added in the basic model, the TIN is not automatically updated. This schema can also not enforce the consistent representations of lines or polygon outlines (e.g., breaklines) in both the basic model and the TIN. All of these operations have to be taken care of by the application software (i.e., by a specific DTM editing module), since the database mechanisms cannot provide the appropriate consistency checks. Another major drawback of this schema is its inefficiency. Since the information pertaining to a single triangle is distributed over three tables, numerous relational joins are necessary when this information is accessed at run time. While the winged triangle structure is not inefficient as an internal data structure, it is slow when used for database storage, slowing down performance by orders of magnitude in comparison to file-based storage.



Vertices

vert_nr	x	y	z	neighbors
1	x1	y1	z1	2, 6, 17, 19
2	x2	y2	z2	1, 19, 22, 25, 3, 4, 6
3	x3	y3	z3	2, 25, 31, 4
4	x4	y4	z4	2, 3, 31, 39, 5, 6
5	x5	y5	z5	4, 39, 42, 43, 6
6	x6	y6	z6	1, 2, 4, 5, 43, 17

Fig. 2: Vertex-oriented data structure

A more compact schema is shown in Figure 2. It is based on a vertex-oriented data structure, needing only one table to fully represent the TIN topology. For each vertex, its adjacent vertices are stored in a counterclockwise order. Thus, a given vertex plus any two consecutive neighbors define a triangle (except for vertices on the hull of the triangulation). Since the number of neighbors for each vertex may vary (but is usually around 5 to 6), the DBMS must provide fields of variable length (bulk data type); this requirement is not met by all commercial RDBMS. This vertex-oriented schema performs much better in terms of database storage and retrieval than the winged triangle approach. On the other hand, it is also not very useful as an internal data structure for surface modeling, and is usually transformed into triangle or edge-oriented representations (Heller 1990). While trans-

formations between different TIN data structures are possible, they slow down the overall performance of the terrain modeling operations.

Many practical DTMs are based on models that are much too large to fit into the memory of any computer. Even if a generously equipped machine could handle the entire model internally, it would be quite inefficient to always read in the entire data set if only a small portion of the model is needed. This is a very realistic scenario as it is most likely that complex models are not built in one run, but in progressive, incremental steps. At each refinement step only a locally confined part is accessed and modified. It is therefore quite important to be able to efficiently access local subsets on the external storage. It must be attempted to structure the model spatially and organise the data on disk according to their spatial configuration. Furthermore, algorithms must be designed so they can profit from the spatial structuring. The data that is typically needed concurrently should be stored correspondingly on disk. Thus, it is possible to substantially reduce the disk-access for frequent operations. Of course, further improvements can be achieved by not storing data that can be reconstructed more efficiently than read. As the research of Heller (1993) has shown, using a hierarchical triangulation scheme on dynamically split Delaunay buckets with a compact, point-based but edge-oriented data structure leads to a very promising solution for disk-based triangulation methods.

TRANSFER OF TERRAIN-RELATED DATA

The portability of DTMs and associated data is a very practical issue. The ease with which the transfer of terrain-related data between different systems can be achieved mainly depends on the complexity of the relations which are modelled by the DTM and on the functionality of the sending and the receiving DTM system. While the transfer of digital cartographic data has received wide attention in the past few years, culminating in the publication of national standards such as the Spatial Data Transfer Standard (SDTS) in the USA (DCDSTF 1988), the transfer of DTMs between different systems can still pose considerable problems.

Because of their simple structure, data transfer can be accomplished relatively easily for gridded DTMs. Grids can be treated like raster data, including additional information about grid spacing, the spatial extent of the DTM, the source of the data and other descriptive information. One possible exchange format that is used by the US Geological Survey for the distribution of their DTM product called 'Digital Elevation Model' (DEM) is described in USGS (1987). SDTS offers a more generic exchange mechanism for grids and rasters.

For the feature data making up the original observations of the basic model, the same data transfer formats can be used as for point, line, and area features basically. However, it is crucial in this case that the third dimension – that is, the z-coordinates of the points and lines – can be transferred as well. While most data transfer formats are focusing on purely planimetric cartographic data, more generic formats such as SDTS can accommodate the transfer of z-coordinates.

For TINs, no published transfer format exists to our knowledge. Since TINs are a more generic structure than grids, various alternative data structures exist, and most DTM software packages use different internal data structures as well as storage structures. In order to transfer the complete information contained in a TIN, the exchange mechanism not only must be capable of transferring the TIN geometry, topology, and descriptive information (e.g., information about data quality). Additionally, a TIN may also be locally adjusted to account for breaklines or contour data, and modified gradients may be inferred to honor surface discontinuities. Finally, the interpolation method that is used is of great importance for determining the actual shape of the modelled surface.

While it is possible to exchange the geometry and descriptive information of a TIN using comprehensive transfer standards such as SDTS, transferring the

specific topological relations of a triangulation can only be accomplished in a rather inefficient way (e.g., by treating triangles as polygons). The only available mechanism of 'transferring' TINs currently is therefore to exchange the original observations between systems and rebuild the TIN topology in the receiving system. This may be feasible if the same form of triangulation (e.g., Delaunay) can be used in both systems. However, if the triangulation has been modified in any way or if the interpolation methods are not equal in both systems it is unlikely that an equivalent TIN can be reconstructed in a different system. Thus, the comparability of results of operations (e.g., visibility analysis or gradient calculation) generated from different DTM systems is hardly possible.

It would be relatively simple to extend standards like SDTS by definitions for TIN topology such as the ones shown in the previous section. It would thus be possible to transfer the TIN geometry and topology. However, as mentioned above, a complete definition of the surface also includes rules for specific local adaptations of the TIN and interpolation methods. A possible solution for the transfer of these procedural elements could be the standardization of construction and interpolation methods. However, this approach would be unrealistic given the wide range of available methods and the diversity of potential uses. A more promising approach is to make use of techniques for the persistent storage of methods as they are available in object-oriented databases. One example is the commercial object-oriented DBMS ObjectStore (Object Design 1991), which allows to implement persistent objects, complete with methods, in the object-oriented programming language C++. The use of common, standardized notations such as C++ may thus provide a mechanism for the exchange not only of data, but also of methods between different GIS.

CONCLUSION

This paper has sought to discuss the requirements and different strategies with respect to functional and database integration of DTMs into GIS. The discussion has shown that while theoretically, integration of DTMs into the DBMS of GIS would have benefits such as increased data security and multi-user access control, these useful properties cannot yet be exploited given the current state of research in spatial databases. Most of the advantages that DBMS can provide for attribute data, such as consistency checking and query operations, must be handled by the application software in the case of spatial data, especially for DTMs. Because database schemata currently used to represent DTMs are inefficient, DBMS currently are not much more than more secure, but also a lot less efficient storage systems in comparison to file-based storage. Presently, DTMs integrated into GIS software systems thus take a practical approach focusing on functional integration: the basic model is stored in the GIS database, and the DTMs are derived thereof and are stored in file structures. As an extension, descriptive information about the DTMs could be stored in the DBMS, allowing queries relating to entire models. As long as tasks are involved that require only little interactivity, modifications to the user interface and query language are only minor.

Nevertheless, the general trend is leading towards better database integration, for planimetric as well as for surface data due to requirements for increased data volumes and interactivity. Future research should foremost focus on the development of more efficient database schemata for DTM data structures. Special consideration should be given to data structures that can be used for disk-based triangulation algorithms (Heller 1993), as they can provide an equivalent representation in memory and secondary storage. Finally, the transfer of DTMs and related data remains a problem, requiring increased attention in future attempts to develop transfer standards that extend beyond cartographic (i.e., planimetric) data.

ACKNOWLEDGEMENTS

I would like to thank my colleague Martin Heller for fruitful discussions and for reviewing the manuscript.

REFERENCES

DCDSTF (Digital Cartographic Data Standards Task Force) 1988, The Proposed Standard for Digital Cartographic Data. *American Cartographer*, **15**(1): 9-140.

Frank, A.U. 1988, Requirements for a Database Management System for a GIS. *Photogrammetric Engineering and Remote Sensing*, **54**(11): 1557-1564.

Fritsch, D. and Pfannenstein, A. 1992, Integration of DTM Data Structures into GIS Data Models. *International Archives of Photogrammetry and Remote Sensing*, **29**(B3): 497-503.

Galyean, T.A. and Hughes, J.F. (1991). Sculpting: An Interactive Volumetric Modeling Technique. *Computer Graphics*. (SIGGRAPH '91), **25**(4): 267-274.

Haas, L.M. and Cody, W.F. 1991, Exploiting Extensible DBMS in Integrated Geographic Information Systems. In: Günther, O., and Schek, H.-J. (eds.), *Advances in Spatial Databases*, Lecture Notes in Computer Science **525**. Berlin: Springer-Verlag, 423-450.

Healey, R.G. 1991, Database Management Systems. In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.). *Geographical Information Systems: Principles and Applications*, London: Longman, **1**: 251-267.

Heller, M. 1990, Triangulation Algorithms for Adaptive Terrain Modeling. Proceedings Fourth International Symposium on Spatial Data Handling. Zurich, Switzerland, **1**: 163-174.

Heller, M. 1993, A Synthesis of Triangulation Algorithms. Technical Report. Department of Geography, University of Zurich, August 1993, 22 pgs.

Höhle, J. 1992, The Object-Oriented Height Model and Its Application. *Int. Archives of Photogrammetry and Remote Sensing*, **29**(B4): 869-873.

Hussey, K.J., Hall, J.R. and Mortensen, R.A. 1986, Image Processing Methods in Two and Three Dimensions Used to Animate Remotely Sensed Data. *Proc. IGARSS '86 Symposium*. Zurich, 8-11 September 1986, **2**: 771-776.

Meyer, B. (1988). Object-Oriented Software Construction. New York: Prentice-Hall.

Molenaar, M. 1990, A Formal Data Structure for Three Dimensional Vector Maps. *Proceedings EGIS '90*. Amsterdam, The Netherlands, **2**: 770-781.

Object Design, Inc. 1991, *ObjectStore User Guide*. Burlington.

Steidler, F., Dupont, C., Funcke, G., Vuattoux, C., and Wyatt, A. 1990, Digital Terrain Models and Their Incorporation in a Database Management System. In: Petrie, G., and Kennie, T.J.M. (eds.), *Terrain Modelling in Surveying and Civil Engineering*. Caithness, UK: Whittles Publishing Services, 209-216.

USGS 1987, Digital Elevation Models. *US Geological Survey Data Users Guide 5*. Reston, VA: USGS.

Waugh, T.C. and Healey, R.G. 1987, The GEOVIEW Design: A Relational Data Base Approach to Geographical Data Handling. *International Journal of Geographical Information Systems*, **1**(2): 101-118.

Weibel, R. and Heller, M. 1990, A Framework for Digital Terrain Modeling. *Proceedings Fourth International Symposium on Spatial Data Handling*. Zurich, Switzerland, **1**: 219-229.

Weibel, R. and Heller, M. 1991, Digital Terrain Modelling. In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.). *Geographical Information Systems: Principles and Applications*, London: Longman, **1**: 269-297.

Issues in Iterative TIN Generation to Support Large Scale Simulations

Michael F. Polis*
David M. McKeown, Jr.
Digital Mapping Laboratory
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

Large scale distributed simulation has been used to support training for battalion level operations within the DoD/Army's SIMNET facility. One of the key components for realistic combat training, through the use of hundreds of low-cost, high realism simulators, all linked using a variety of telecommunications technologies, is the fidelity of the underlying shared environment. For ground-based simulations the environment includes the terrain, road networks, buildings, and natural features such as drainage, forests, and surface vegetation. Given the limitations in graphics rendering capabilities for the low-cost computer image generation systems associated with SIMNET, one must trade fidelity in terrain representation for feasibility of simulation.

Our work has focused on reducing the number of polygons that must be rendered in a real time simulation while improving the overall fidelity of the terrain visualization. In this paper we present a new method for the generation of a terrain representation beginning with a digital elevation model (DEM) and producing a triangular irregular network (TIN) suitable for scene generation. We compare our technique to two commonly used methods, VIP and LATTICETIN and provide a performance analysis in terms of residual error in the TIN, and in the number of points required by each method to achieve the same level of fidelity. We also briefly outline the use of selective fidelity to improve the level of detail in selected sub-areas of the DEM.

While motivated by the constraints of real-time distributed simulation, this research has applications in many areas where an irregular network is a more computationally efficient representation than a dense digital elevation model.

INTRODUCTION

The Triangular Irregular Network (TIN) has distinct advantages over the more traditional Digital Elevation Model (DEM) grid for many applications such as visualization and simulation. The irregular point placement of a TIN permits the resolution of the model to adapt to terrain complexity. Many computations are simplified by the polygonal structure, such as drainage modeling and computer image rendering.

We first discuss previous approaches to TIN generation. Then we present a new method for generating a TIN from a DEM. The method is iterative, with the initial TIN containing the four corner points of the original DEM. At each step, an approximate DEM is interpolated from the TIN and an error surface is generated. We add correction points by detecting maximal errors in this surface and generate a new triangulation. The TIN is evaluated at the end of each step by a user defined stopping criterion, such as

*This research was primarily sponsored by the U.S. Army Topographic Engineering Center under Contracts DACA72-87-C-0001 and DACA76-91-C-0014 and partially supported by the Avionics Lab, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, Arpa Order No. 7597. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Topographic Engineering Center, or the Advanced Research Projects Agency, or of the United States Government.

point count or maximum global/RMS error.

Since our algorithm generates a series of approximations to the DEM we can generate a hierarchy of triangulations simply by storing intermediate resolution TINs as they are generated. This hierarchy provides multiple levels of detail useful in visualization and modeling. A large area can be broken into tiles, with successively lower levels of detail for tiles successively more distant from the viewer. This allows the display of a large area in a limited polygon budget.

After describing our algorithm, we compare our method with two commonly used techniques, VIP [2] and LATTICETIN. We then discuss pragmatic issues in the use of our triangulation method for computer simulation and limitations imposed by computer image generation using a large real world terrain database.

PREVIOUS WORK

Interest in the generation and utilization of irregular terrain representations dates back nearly twenty years. The work of Peucker [6, 7] introduced the TIN terminology and outlined the basic TIN construction problem in terms of sampling constraints, as well as manual and automated techniques for triangulation. They also described the importance of topological structure in selecting points for inclusion in the TIN. Over the years a variety of techniques for manual and automatic point selection have been proposed and implemented.

Traditionally, points have been selected from stereo imagery by a skilled operator [1]. Recently, work has been done on automatic TIN generation from other terrain models already rendered in digital form. For example, Chen and Guevara [2] generate a TIN from raster data, and Christensen [3] generates a TIN from contours on a digitized topographic map.

One of the earliest manual point selection methods is the one used in the ADAPT system. This system, mentioned in Peucker [6], uses man-machine interaction to aid the user in producing a TIN which accurately represents the original terrain. The system goes through an iteration process in which it indicates inadequate parts of the TIN, which the human operator then improves.

Fowler and Little [5] select points in two stages. First, DEM points are automatically classified as peaks, pits, passes, etc. "Surface specific" points are selected based on this classification. A three-dimensional extension of a line simplification algorithm is applied to the points of ridge and valley lines. Additional support points are then selected adaptively until error is within a user defined tolerance. The complexity of implementing this algorithm has prevented its widespread use.

DeFloriani et al. [4] present a hierarchical triangulation method and data structure. With Hierarchical Triangulation (HT) the area is initially divided into triangles. There are many variants of this scheme, but typically, a rectangular area will only be divided into two triangles. Each triangle can then be subdivided by adding a point interior to the triangle and connecting this point to the triangle vertices. Typical criteria for this are: triangles are subdivided only when their maximum error exceeds a tolerance; and the new point is selected by testing all points, and using the one which gives the most improvement. This produces a hierarchy of approximations to the DEM, where later members of the hierarchy include all the points and triangle edges of earlier members. Sharing triangle edges is useful for operations like point location, but it is undesirable in simulation. First, error along shared edges cannot be improved, and these lines will be visible in the simulated terrain. Second, never dividing edges means that the number of sliver triangles is very high in later TINs. Finally, note that DEM boundaries are triangle edges, and can therefore never be improved. All these factors lead to artificial looking terrain.

Chen and Guevara [2] present an automatic point selection procedure called VIP for selecting points directly from the DEM. VIP is essentially a high-pass filter which gives an importance value based on the distance a DEM point is from the 4 lines connecting its diametrically opposed neighbors. This would seem to be a good solution to the point selection problem; all points in the DEM are ordered in terms of importance, which permits rapid selection of a point set of any given size. The procedure does have several problems which stem from the local nature of the importance criterion. First, the peak of

a small, sharp hill will be considered more significant than the peak of one which is large, yet slopes gently. Second, the VIP procedure chooses nearly all the points of valleys and ridge lines. This is desirable in the sense that these are important features to capture. However, this is clearly unnecessary in those places where the ridge or valley follows a straight line, since we can represent any straight line with two points. Both of these problems arise because only the 8 neighboring points of any DEM point are considered. An additional disadvantage is that the filter is undefined at the DEM boundaries. The definition can be extended, or another point selection method can be used for the boundaries, but some special handling is required.

Our algorithm, like HT and Fowler and Little, improves the TIN until it meets user defined criteria. Unlike Fowler and Little, the algorithm is simple, and point addition begins with a simple TIN containing only 4 corner points, although the algorithm could also begin with "surface specific" points. Like HT, we produce a hierarchy of triangulations, but in this hierarchy, only points are shared allowing error to be improved everywhere. Unlike both HT and VIP, the boundary is approximated in the same manner and to the same accuracy as the rest of the DEM.

ITERATIVE TIN GENERATION

We begin with a set of initial points. This always includes the corners of the DEM, and may include other points which the user wants to have included in the final TIN. A Delaunay triangulation [9] of these points provides the initial approximation. The TIN produced can be interpolated at every point of the DEM to produce an approximate DEM. Subtracting the original and approximate DEMs gives an error at every point. We find the maximum absolute value of the error and select points with this error value as candidate correction points. We then add the candidate points to the TIN if they are not within a specified vertical and horizontal tolerance of any point already in the TIN. The values used are 300m for the horizontal tolerance and 20m for the vertical tolerance. This step simply makes more efficient use of the point budget by keeping simultaneous corrections apart. The algorithm functions tolerably without it. Hence the precise values are not critical, but the values given work well for a wide variety of terrain. If all candidate points fail this criterion, then they are only tested against each other, so that at least the first will be added. The points are again triangulated. The process of selecting and adding points can be repeated until some stopping criteria such as RMS error, point count, or maximum error is met.

This TIN generation method is very flexible. The initial point set can include points from a variety of sources. These may be points selected manually by the user, boundary points required to match an adjacent terrain model, or the points selected by another TIN method. This means that our algorithm can be used to improve any existing TIN, such as one generated manually or by selecting peak and ridge points.

The point selection process can be modified to use an importance mask. This gives an importance value for each DEM location, which the error at that point is multiplied by before points are selected. High importance can be assigned to a region to concentrate points there. Individual points or entire regions can be excluded from point selection by setting an importance of zero.

We believe that this method makes very effective use of its points. Every point is added to correct a specific error, and the adjacency criterion applied to the candidate points usually ensures that only one point is used to correct the error.

In comparison to our previous algorithm based on the contouring of the error DEM at each iteration [8], the present one is greatly simplified. The iterative approach is retained, but fewer points are selected each iteration. The increase in the number of iterations requires a faster triangulation. Our previous method used a greedy triangulation which was based on the error measured along a candidate edge rather than its length. By considering the fit to the underlying terrain, superior triangulations were produced. The Delaunay triangulation tries to produce a triangulation with minimum total edge length and equilateral triangles, but does not consider the underlying terrain. It is, on the other hand, very fast, even when triangulating a large number of points, and it can add points incrementally. Its speed makes selecting only a few points per iteration practical, and the improvement in point quality compensates for the loss in triangulation quality, resulting in a better overall algorithm.

As the TIN generation process proceeds, the algorithm generates a hierarchy of triangulations. Any number of these can be saved to provide multiple levels of detail. The hierarchy will share points, but not triangle edges, unlike HT. This is desirable because not sharing edges means that error along edges in a triangulation can be improved. Hierarchies like that of HT also tend to include many long thin triangles which poorly represent the terrain and give it an artificial appearance during visualization.

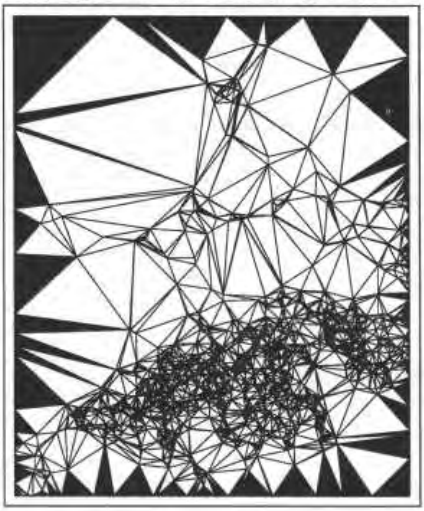


Figure 1: VIP 2.891% TIN

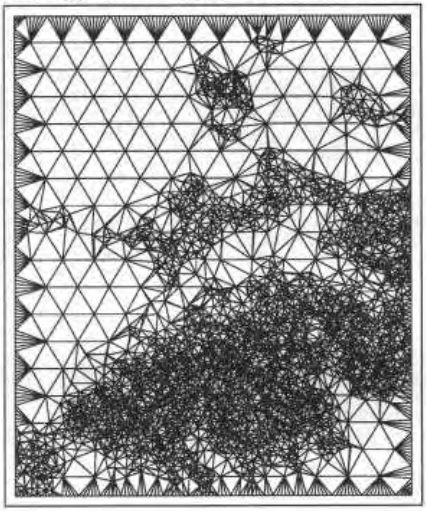


Figure 2: 7 iteration LATTICETIN

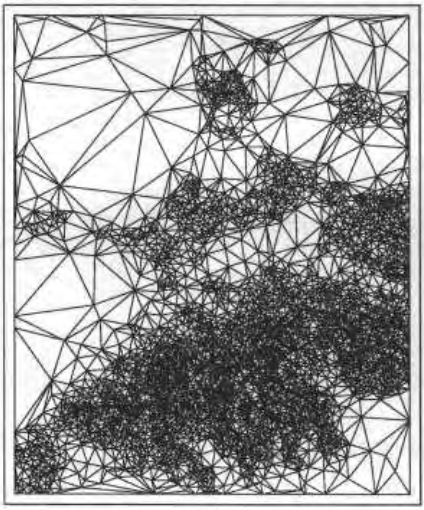


Figure 3: 5000 point Iterative TIN

Figure 4: Relief of Mt. Tiefort DEM

Figure 5: VIP 2.891% TIN

Figure 6: 7 iteration LATTICETIN

Figure 7: 5000 point Iterative TIN

Figure 8: Mt. Tiefort DEM

PERFORMANCE ANALYSIS OF TIN GENERATION METHODS

In this section we compare the performance of our algorithm to two commercially available algorithms. The DEM used in Figure 4 is of Tiefort Mountain, which is located in the California desert in the National Training Center.* The DEM is composed of Defense Mapping Agency DTED Level 2 data having a 30 meter UTM grid spacing, and containing 172,960 points. The TIN methods will be compared by having each method select points from the DEM for two TINs, of about 2500 and 5000 points. The 5000 point TINs are compared visually and all the TINs are compared statistically.

We compared our algorithm to ESRI's tools for TIN generation in ARC/INFO. There are three TIN generation methods included: HIGHLOW, VIP, and LATTICETIN. HIGHLOW selects local minima and maxima. Triangulating these produces a TIN bounded by a convex polygon smaller than the DEM border. Since HIGHLOW does not generate a TIN which covers the whole DEM, its accuracy cannot be properly compared with the other methods, especially since the ability to represent the DEM boundary is important. The VIP implementation deals with the undefined boundary values by giving boundary points a higher importance than any other points. Thus they are included in the TIN regardless

*Tiefort Mountain is the tallest mountain in the 50km by 50km DEM described in Section 6.

of the percentage of points selected. The TIN shown in Figure 1 was generated with VIP using 2.891% of the DEM (5000 points). A lower resolution TIN was produced using 1.4455% of the TIN (2500 points). LATTICETIN starts with regularly spaced boundary and interior points and then iteratively adds additional points, apparently* by adding points inside triangles which do not meet an error tolerance. After running for four iterations, a TIN of 2794 points was produced. Continuing to the seventh iteration produced the 5188 point TIN shown in Figure 2.

Our algorithm produced the TIN shown in Figure 3 having 5000 points selected from the DEM and a 2500 point TIN for statistical comparison. Boundary points were selected in the exactly the same manner as interior points. The difference is apparent when Figures 1 through 3 are compared. By choosing all (or even a large fraction) of the edge points, both VIP and LATTICETIN give an unnatural representation near the boundary. LATTICETIN also has the flaw of beginning with a regular arrangement of points. This reduces its flexibility, and causes it to perform poorly with a small point budget. The VIP TIN has characteristically selected nearly all its points along sharp ridge and valley lines. This has two effects. First of all, the ridges are not represented efficiently, leaving little of the point budget to represent anything else. Second, since much of the low area is smooth, no points are selected there, and the TIN connects ridgelines.

Error Analysis

The accuracy of the TINs are shown in Table 1. Errors are given as elevation differences in meters. The column labeled *equivalent points* shows the number of points the iterative algorithm took to have a lower RMS error than the listed method. VIP is done some injustice here since it is forced to include all the boundary points, but there are only 1668 of these, and even excluding these points, the iterative algorithm outperforms VIP in both cases. The error figures for the 5000 point LATTICETIN are close to our method, but this is deceptive. Error improves slowly with this many points, so the difference in RMS error of less than a meter represents well over a thousand points. Both LATTICETINs trail by nearly the same number of points.** This is evidence of the efficiency of our method.

Performance Results for Mt. Tiefort DEM				
Method	TIN Points	RMS Error	Maximum Error	Equivalent Points
VIP	2500	48.5342	216	90
LATTICETIN	2794	6.25136	81	1332
Iterative	2500	4.08376	28	N/A
VIP	5000	36.2015	143	107
LATTICETIN	5188	2.94568	31	3719
Iterative	5000	2.26767	19	N/A

Table 1: Numerical Accuracy

Comparison of Static Visualization

A more subjective comparison can be made by using each of the three TIN methods to construct a perspective view of the terrain. While dynamic motion in a real-time simulation may tend to blur detail and lesson the effect of these static views, such a comparison is still worthwhile. In Figures 5 through 8 we show the DEM and each of the TINs from an near ground view looking south. VIP has not selected any points in the area between the near ridgeline and Tiefort Mountain, so the triangulator has directly

*The documentation I have seen does not explain the algorithm well.

**1462 for the 2794 point TIN and 1469 for the 5188 point TIN

connected the two. Both LATTICETIN and our iterative algorithm have approximated the terrain well, although there are some details which are better defined in our TIN. Several minor ridges are visible only in the iterative TIN.

Figures 9 through 12 show a near vertical view looking from the south at Tiefort Mountain and the pass just to its east. Once again, VIP has selected few points in the low-lying areas. Both LATTICETIN and our algorithm capture the terrain well, but our algorithm captures slightly more detail. The ability to capture additional detail can be very valuable in ground vehicle simulation, where micro-terrain provides cover in combat.

Figure 9: VIP 2.891% TIN

Figure 10: 7 iteration LATTICETIN

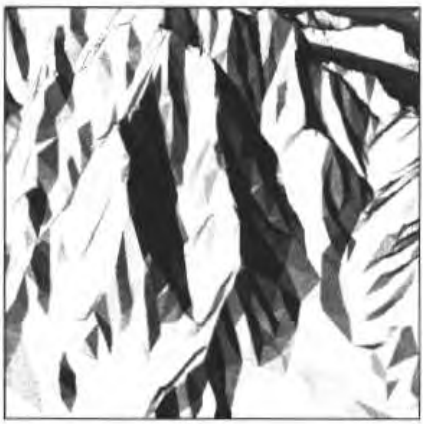


Figure 11: 5000 point Iterative TIN

Figure 12: Mt. Tiefort DEM

GENERATION OF LARGE-SCALE TINS FOR SIMULATION

A digital elevation model shown in Figure 13 was constructed to support a SIMNET training exercise. The original DEM, provided to us by the U.S. Army Topographic Engineering Center (USATEC), covers an area 50 kilometers on a side (2500 square km) including the National Training Center (NTC), Fort Irwin, California. The range of elevations in the DEM is approximately 1500 meters. This area is primarily desert, with some highly eroded mountainous areas and intricate alluvial fans running to the desert floor. The sheer size of the area presents significant problems. The DEM consists of 1979×1979 points, nearly 4 million elevation posts. To maintain the desired polygon density for the SIMNET computer image generation systems, only 76,500 points (less than 2%) were to be selected for the TIN.

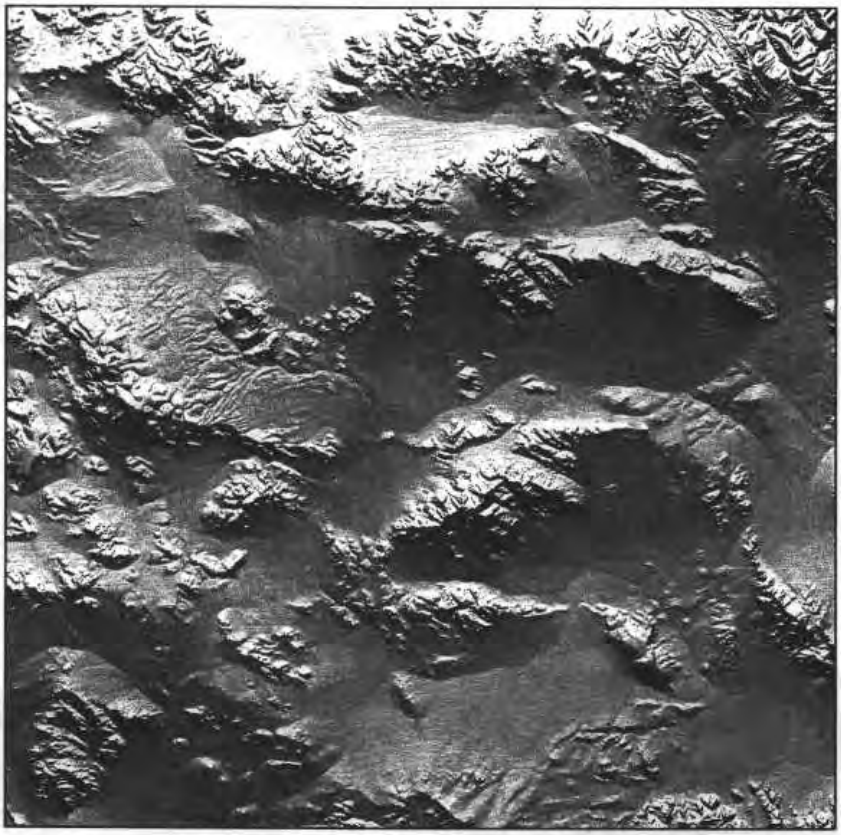


Figure 13: Shaded Relief of NTC 50×50km DEM

SIMNET Terrain Model

The usual SIMNET terrain model is a triangulated 125m DEM. The hardware is capable of using any polygonal terrain model, but a DEM has been used for ease of generation. The terrain and cultural features are divided into 500m square load modules. While the simulator is running, a block of load modules around the observer's position are loaded and displayed. This prevents the gaming area from being limited in size by hardware rendering capabilities. Polygons are not allowed to cross load module boundaries. If they do, they must be clipped at the boundary, which increases the total number of polygons. The 125m spacing of points in the original model was chosen to avoid this problem; load module boundaries fall along triangle edges, so no clipping is necessary. When a TIN is used for the terrain model, clipping is necessary, and given the small load module size, will increase the number of polygons markedly, although some of the additional points can be useful. At the load module corners, where the clipping lines intersect, we know points will be added. These points can be added to the initial TIN, and give a higher quality TIN in the same effective point budget. Another possible improvement to TIN accuracy would be to use elevations interpolated from the DEM for the points which result from clipping. This was not done. Instead, triangles which resulted from clipping were kept coplanar with the original triangle. By making the load module boundaries less apparent, this made the terrain less artificial in appearance.

The point budget is based on a number of factors. We want the same average polygon density as the 125m DEM, 32 per load module, 320,000 in the whole TIN. This bound should apply after clipping, which has a somewhat unpredictable effect. Also, certain

local limits must not be exceeded. Load modules cannot contain more than 100 polygons total; excess polygons must be put into a limited global pool. If there are too many polygons in the field of view of the simulator, it will not be able to display all of them. It was assumed that the latter constraint would be satisfied if no 4km square contained over 4096 triangles. All these factors suggested that the point budget should be chosen conservatively. Each load module in a 125m DEM has 9 interior points, 4 corner points and 12 other edge points. The point budget was 9 points per load module, 90,000 total, plus corner points. This violated the constraints, so the budget was further reduced 15% to 76,500 points. Points around the border of the DEM spaced 125m apart were also added so that the load modules would mesh with surrounding DEM data. The other border points were marked as unavailable by using a mask with their importance set to 0.

Selective Fidelity

An additional complication for the TIN generation process was the desire for reduced fidelity in the mountainous areas to permit increased detail in the areas of alluvial fans and on the desert floor. This was primarily driven by the fact that mountainous areas are not accessible to ground vehicles (simulated or otherwise); yet, due to their height and complexity, they tend to accumulate a large number of TIN points. This decreases the budget available for other areas of the terrain. An overlay indicating the mountainous areas was provided by USATEC, and this provided importance values in the mask. Importance for the low lying areas was set at 8 times that in the mountains. We smoothed the importance grid with a 7x7 Gaussian filter to avoid problems that might result from a discontinuity at the boundary of the mountainous area.

Computational Issues

The NTC test area is also large enough that generating the TIN for the whole area at once is impractical. Since there are local constraints on point density, this makes it doubly useful to divide the area into tiles which will each have a TIN generated separately. The area was divided into 36 tiles. This brings up the issue of consistency along the borders between tiles. A variety of approaches are available; points could be spaced regularly at 125m intervals, or they could be selected by a 1D line approximation algorithm. In both cases, an arbitrary decision as to the number of points along the border would have to be made. Instead, each TIN was generated by starting with the border points of adjacent TINs which already existed, allowing it to select border points naturally along the remaining borders. This permits the generation of separate tiles in parallel as long as adjacent TINs are not generated at the same time. Three DECStation 5000/200s using a common network file system worked in parallel to generate the TINs in 6 hours. The TINs were combined by re-triangulating, which made the tile boundaries invisible.

Final Results

Figure 14 shows a shaded relief terrain representation of the TIN produced by our method. The TIN was generated using selective fidelity in the mountainous areas. Using less than 2% of the original DEM points we were able to construct a TIN which shows an RMS elevation error of under 3 meters when compared to the original DEM. From a qualitative standpoint it appears that the major topographic features are generally preserved and that detail in the alluvial fans and desert floor areas are also quite good. This impression was confirmed using the SIMNET system at USATEC and driving an M1 tank (simulated) through the terrain.

Table 2 presents statistics about the TIN. The DEM actually covers an area slightly larger than 50×50km. TIN points were not selected in this area, and it is ignored in the statistical analysis. The ridges and mountains were represented well enough to allow their use for navigation, while the low lying areas were represented at a high level of detail. In many places, the drainage and alluvial fans are almost indistinguishable from the original.

The model is a significant improvement over the original 125m DEM in terms of function as well as appearance. Scout vehicles usually seek cover in drainage features. Since these features were small, they were smoothed over in the 125m DEM. The TIN was able to preserve these features, making the simulation more realistic.

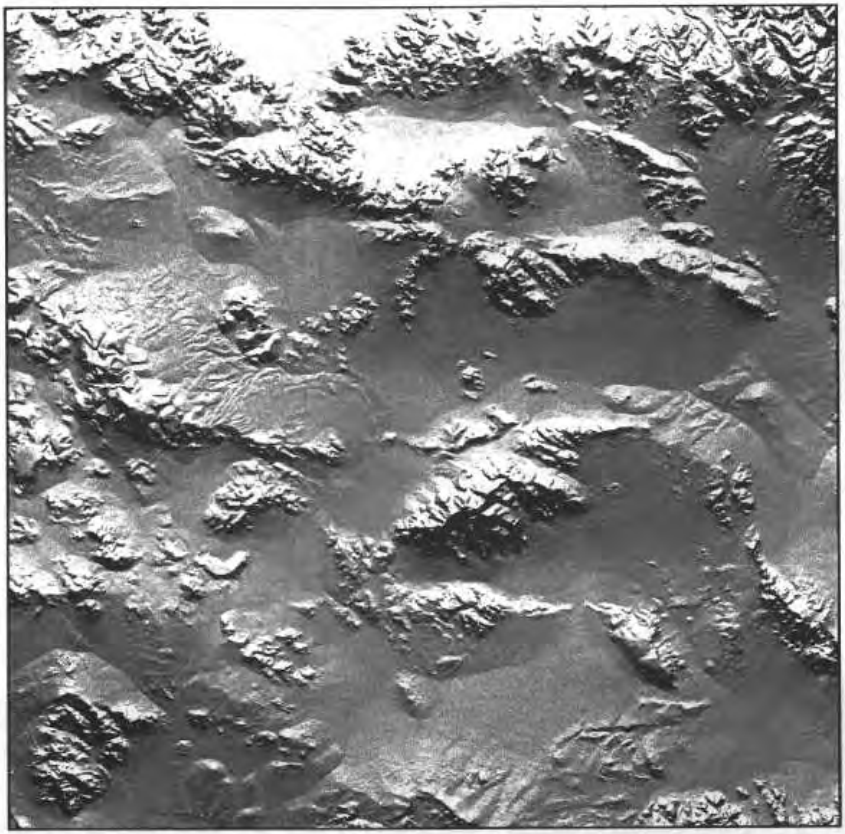


Figure 14: Shaded relief of NTC TIN

Performance Results for Mt. Tiefort DEM				
Area	Mask Points	TIN Points	RMS Error	Maximum Error
Mounts	1,313,410	28,532	5.26791	52
Plains	2,587,215	64,939	0.86912	24
Overall	3,900,625	93,471	2.99929	52

Table 2: Numerical Accuracy

CONCLUSIONS

In this paper we have described a new iterative TIN generation method and compared its results to two popular methods. Our method has several advantages: it is simple and flexible, it generates a hierarchy of TINs, and it makes efficient use of its point budget. We also demonstrated the ability of our method to construct a large-scale TIN with improved terrain fidelity to support a SIMNET training exercise. Future work will focus on the integration of road and drainage networks compiled from high resolution digital maps into the coarse resolution TIN. We will continue to exploring the use of selective fidelity to preserve fine terrain detail.

ACKNOWLEDGEMENTS

We thank Mr. Doug Caldwell of the Autonomous Technologies Division, Research Institute, U.S. Army Topographic Engineering Center (USATEC), Fort Belvoir, VA for his help in describing the selective fidelity terrain generation problem and in generating the VIP and LATTICETIN examples used in this paper. Ms. Linda Graff of USATEC produced the mountainous area overlay importance grid used for the NTC example. This paper benefitted from careful readings by Jeff Shufelt and Steve Gifford. Our colleagues in the Digital Mapping Laboratory maintained a high level of humor as their workstations were commandeered for the NTC TIN construction experiment.

REFERENCES

[1] J.R. Carter.
Digital Representations of Topographic Surfaces.
Photogrammetric Engineering and Remote Sensing 54(11):1577-1580, November, 1988.

[2] Zi-Tan Chen and J. Armando Guevara.
Systematic Selection of Very Important Points (VIP) from Digital Terrain Model for Constructing Triangular Irregular Networks.
In N.R. Chrisman (editor), *Proceedings of the Eighth International Symposium on Computer-Assisted Cartography*, pages 50-54. 1987.

[3] A. Christensen.
Fitting a Triangulation to Contour Lines.
In N.R. Chrisman (editor), *Proceedings of the Eighth International Symposium on Computer-Assisted Cartography*, pages 57-67. 1987.

[4] L. DeFlorani, B. Falcidieno, and C. Pienovi.
Structured Graph Representation of a Hierarchical Triangulation.
Computer Vision, Graphics, and Image Processing 45:215-226, 1989.

[5] R.J. Fowler and J.J. Little.
Automatic Extraction of Irregular Network Digital Terrain Models.
In *SIGGRAPH*, pages 199-207. 1979.

[6] T.K. Peucker, R.J. Fowler, J.J. Little, and D.M. Mark.
Digital Representation of Three Dimensional Surfaces By Triangulated Irregular Networks (TIN).
Technical Report 10, Office of Naval Research, Geography Programs, 1976.

[7] T.K. Peucker, R.J. Fowler, J.J. Little, and D.M. Mark.
The Triangulated Irregular Network.
In *Proceedings of the ASP Digital Terrain Model Symposium.* 1978.

[8] M. Polis and D. McKeown.
Iterative TIN Generation from Digital Elevation Models.
In *Proceedings of the DARPA Image Understanding Workshop*, pages 885-897. Morgan Kaufmann Publishers, Inc., San Mateo, CA, January, 1992.

[9] F.P. Preparata and M.I. Shamos.
Computational Geometry.
Springer-Verlag, 1985.

AN INTEGRATED DTM-GIS DATA STRUCTURE: A RELATIONAL APPROACH

by
M. Pilouk, K. Tempfli
Department of GeoInformatics
ITC
P.O.Box 6, 7500 AA Enschede
The Netherlands

ABSTRACT

To achieve a better solution to utilizing and maintaining both DTM and GIS datasets more efficiently at one time, a new approach to their integration into a unified data structure is investigated using the triangular irregular network and single-valued vector map models. A relational database approach is chosen for the implementation, which demonstrates the practicability of the model with respect to various topological queries and to obtain information about terrain relief from the unified dataset.

INTRODUCTION

The increasing demand for reliable and comprehensive digital data for various purposes governs the need for the integration of diverse geographic information (GI). The integration of DTMs and GISs is one of the issues in this context. A DTM represents terrain relief, thus the shape of the (topographic) surface in 3-dimensional space, whereas a GIS refers to other features of the terrain such as hydrography, soil, land use, etc, which are traditionally presented in 2-D. At present, DTMs and GISs are usually integrated only by interfacing approaches (e.g., Ebner et al, 1990; Mark et al, 1989; Arc/Info, 1991). In such approaches the data structures of the DTM and the GIS remain unchanged, which implies separate data storage and manipulation of the two systems. This can cause problems of integrity of terrain representation, reliability of derived height and slope information, and low efficiency when both sets of information have to be used to facilitate better spatial analysis and graphic presentation. We are therefore investigating a different approach which aims at a higher level of integration. In developing a unified data structure, the paper outlines the basic concept, the data model and the construction of a relational database structure. To demonstrate some queries, the tables were implemented in dBASE IV SQL. A thorough analysis of the query space as well as rules for data editing and updating are still under investigation and therefore not included here.

TRIANGULATION AND SINGLE VALUED VECTOR MAP

Two main principles in structuring data are proximal ordering and decomposition into primitives. Digital terrain relief modelling requires interpolation, which in turn requires that the proximal relationships among the given points are known. As DTMs in most cases are based on nongridded data, adjacency can be expressed best by a triangulation. Triangulation of surveyed points in order to interpolate contour lines for topographic maps was in fact applied long before the advent of computers. Thiessen (1911) polygonal tessellation and its geometric dual, the Delaunay triangulation (see Pilouk and Tempfli, 1992), are of special interest as it establishes a natural neighbour structure. Furthermore, the triangular irregular network (TIN) structure can readily accommodate skeleton data (ridge lines and drainage lines) as triangle edges, which generally increases the fidelity of surface representation (see Pilouk, 1992; Roushannejad, 1993). Among the various advantages of a triangular tessellation, as pointed out frequently in the literature, it is worth mentioning that it allows for local editing and updating of elevation data without elaborate reinterpolation as is necessary in a grid-based structure (see Fritsch and Pfannenstein, 1992).

The concept of triangulation can be applied also to any other vector-structured GI. Any area feature can be decomposed into triangles, a line feature can be decomposed into a sequence of edges of triangles, and a set of point features can be vertices of triangles. This way a congruency can be established between structuring "DTM data" and "GIS data". Molenaar (1988) suggested a model for vector-structured representations of a terrain situation. What he called the formal data structure (FDS) of a single-valued vector map (SVVM) distinguishes three data types: terrain features, their geometric primitives, and their thematic attributes. These elementary data types and the elementary links between them can be graphically represented as indicated in figure 1. The FDS provides a 2-D topology; the coordinates linked to 'node', however, can be 3-dimensional. Nevertheless, interpolation (i.e., surface construction) in a SVVM would be difficult without an adequate tessellation. On the other hand, using a complete 3-D model is expected to cause too much overhead for applications dealing with only the (topographic) surface rather than with phenomena above or below it.

DATA MODEL FOR THE INTEGRATED DTM-GIS

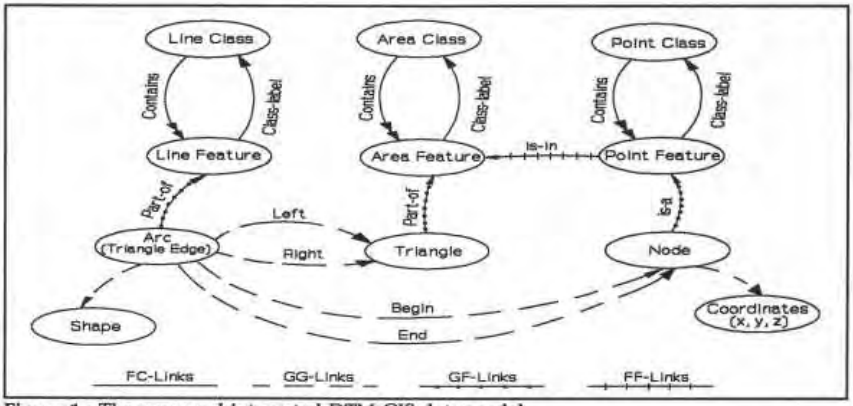


Figure 1 : The proposed integrated DTM-GIS data model

By extending the FDS of SVVM to include on the geometric level the item 'triangle' (see figure 1), a data model is obtained which serves the purpose of unified handling of all surface related data. In the unified structure (UNS), an area feature is not anymore linked to 'arc' directly but to its geometric primitive 'triangle'. An area feature consists of one or more triangles. Arcs are linked to triangles through the 'left' and 'right' geometry-geometry links (GG-links). The vertices of a triangle can be found through the arc-triangle and arc-node links. Each node is represented by one coordinate triple which consists of one X, one Y, and one Z. Since all nodes are equipped with 3-D coordinates, a plane equation can be derived from the three vertices of a triangle. This allows further derivation of information which relates to terrain relief, such as elevation at any point, slope, and aspect. Additionally, the real-time visualization is no longer limited to orthogonal views (the traditional map). Popular visualizations, such as perspectives, pseudo-perspectives, and stereo views, can be generated more efficiently, and can be combined with a cursor to pick-up information from the "3-D graphics".

TOPOLOGICAL ASPECT OF THE INTEGRATED DTM-GIS DATA MODEL

The UNS fits nicely into topological reasoning (see Egenhofer et al, 1989). It consists of 0-simplices (points), 1-simplices (arcs), and 2-simplices (triangles), which are the smallest data elements in 0, 1, and 2 dimensions respectively. This approach means that the decomposition process of 2-cells (area features) into 2-cell simplicial complexes, which are the collection of 2-simplices, is needed. By equipping each 0-simplex with 3-D coordinates (x,y,z), mapping of this model in 3-D metric space becomes possible and

meaningful. The simplicial complexes (e.g. point, line and polygon objects) can be formed from these simplices. Consequently, various topological operations and derivation of topological relationships can be easily performed by using the basic binary relationships (see Egenhofer et al, 1989), because all objects are said to be decomposed into minimal spatial objects.

A RELATIONAL DATA STRUCTURE OF INTEGRATED DTM-GIS

In general, the proposed data model can be mapped into different kinds of data structures (e.g., network, relational, object-oriented, etc.). We have chosen the relational one, considering ease of implementation, flexibility, and availability of various database management systems (DBMSs). The network data structure is powerful but it takes more effort in implementation, while an object-oriented one is even more difficult to implement because it needs careful definition for not only each object but also the methods to access that object. Moreover, only few object-oriented database management systems are commercially available, while a network structure normally has to be implemented by programming. Another reason for choosing a relational approach is its maturity in providing a rigorous procedure for mapping the data model to a data structure. This is known as normalisation, which is the mechanism to ensure data integrity of the database against updating anomalies. The common approach is to decompose a preliminary table into first, second, third, fourth, and fifth normal forms; this way several normalised tables are obtained. However, this seems to be a very tedious task. A more appealing approach has been proposed by Smith (1985), where tables are composed from a dependency diagram. If the procedure is followed correctly, the obtained database tables are then fully normalized. The steps can be summarized in four phases: constructing dependency statement, mapping from dependency statements to dependency diagram, composing relational tables from the dependency diagram, and improving the handling of the relational table by introducing a surrogate key if necessary.

Constructing dependency statement

In this phase the data fields to be stored in the database have to be identified first. Looking at the data model in figure 1, data fields are encompassed by ellipses, whereas the relationships are the labels on the lines connecting two ellipses. The relationships between each pair of fields is analyzed and then translated into a dependency statement. The list of dependency statements is given below. It deviates from the list given by Bouloucos (1990) according to the change that was introduced to the FDS. Modified statements are marked with '*'; added ones with '+'.

(1) A line feature identified by an LID belongs to one LCLASS line feature class.

(2) An area feature identified by an AID belongs to one ACLASS area feature class.

(3) A point feature identified by a PID belongs to one PCLASS point feature class, is represented by one PNODE node number and may fall inside one PAID area feature.

(*4) An arc identified by ARCNR has one BEG starting node and one END ending node.

(*5) Each NODENR node has a position given by a one X x-coordinate, one Y y-coordinate, and one Z z-coordinate.

(6) The original dependency statement on crossing of line features is deleted since line crossings (above/below) are taken care of by given 3-D information and dependency statement 8.

(+7) A triangle identified by TRINR represents at most one TAID area feature.

(+8) An ARCNR arc has at most one LTRI triangle on its left side, at most one RTRI triangle on its right side (null value of LTRI and/or RTRI indicates that this arc is a part of crossing over or under of that part of the surface and thus is not a triangle edge).

(+9) An ARCNR arc represents at most one ALID line feature.

Mapping from dependency statements into dependency diagram

From the above list of dependency statements, the corresponding dependency diagram can be drawn as shown in figure 2. The attributes (data fields) are shown within bubbles. A line between two bubbles indicates a relationship between one data field and the other. A single-headed arrow indicates that it is a single-valued dependency and a double-

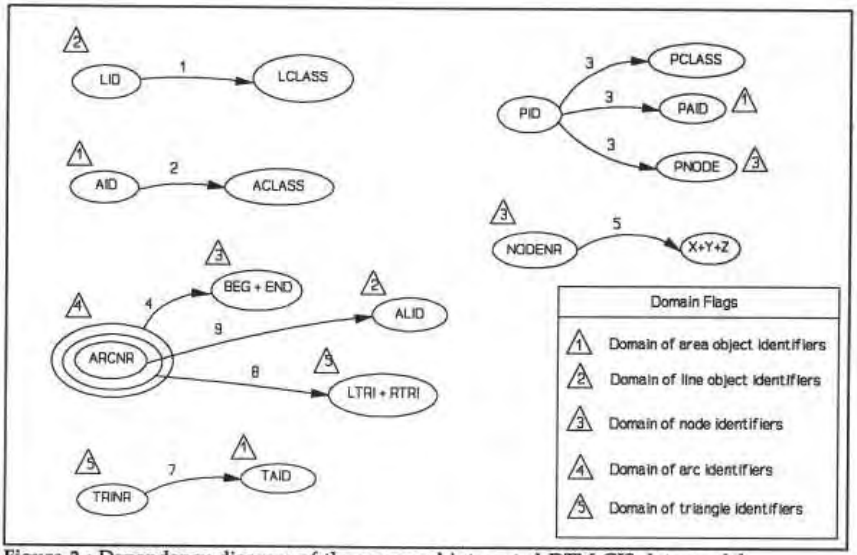


Figure 2 : Dependency diagram of the proposed integrated DTM-GIS data model

headed arrow indicates a multi-valued dependency. More than 1 bubble covering a data field indicates that not all the relationships may apply to every value of the field. For example, ARCNR = 60 in figure 4 has a left and a right triangle but is not part of a line feature. A number adjacent to a line between two bubbles indicates the dependency statement number. Each domain flag which is the indicator of different named fields having a common field type (e.g., TRINR, LTRI, and RTRI are of the same field type representing triangle identifiers) is shown as a number within a small triangle.

Composing relational tables from dependency diagram

First tables are composed from the single-valued dependencies and then from the multi-valued dependencies. A bubble which has no arrow pointing to it becomes a primary key field of one table. A target bubble becomes a data field in the same table. A bubble which is pointed to by an arrow and has a domain flag also becomes a foreign key field in the same table. In case of multi-valued dependency, all data fields emanating arrows comprise the primary key. Special care should be taken here if there are more than three fields comprising a primary key; the table may not be practicable, which results in bad response time. The solution is to split into two tables by introducing a surrogate key acting as the primary key for one table and as a foreign key in the other. The following tables result:

R1: LINE (LID, lclass)
R2: AREA (AID, aclass)
R3: POINT (PID, pclass, paid, pnode)
R4: ARC (ARCNR, beg, end)
R5: NODE (NODENR, x, y, z)
R7: TRIAREA (TRINR, taid)
R8: ARCTRI (ARCNR, ltri, rtri)
R9: ARCLINE (ARCNR, alid)

QUERY DEMONSTRATION

Preparation

For ease of comparison with the SVVM, the example given by Bouloucos et al (1990) and shown in figure 3 is used as a given vector map. Height information is obtained from a DTM which was derived from a contour map. Before integrating the DTM and the vector map, the topological relationships (tables) of the vector map must be constructed. This is assumed to be completed already. The new database is created in the following steps:

(1) Converting all nodes of the map (figure 3) to raster format according to their x and y coordinates. The ILWIS version 1.3 POINTS module was used for this purpose. The

output is the raster image of all nodes. The pixel value of each raster node is the NODENR (from table R5 in the preceeding section).

(2) Overlaying the image obtained from step (1) with the DTM to provide the z-coordinate for each node.

(3) Tessellating the nodes image by using the INTTHS program (see Pilouk, 1992). The output is the Thiessen polygon image, with each polygon given the colour value corresponding to the value of NODENR.

(4) Generating TIN data structure by using the THSTIN program (see Pilouk, 1992). The output is a vertex-based TIN structure. The graphical representation of this result is shown in figure 5.

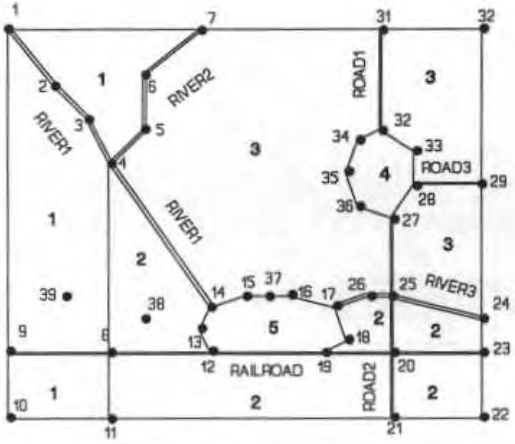


Figure 3 : Single valued vector map (see Appendix A for legend)

(5) Creating the TRIAREA table (R7 in preceeding section) in dBASE IV and then importing the TIN data from step (4) into this table, updating the TAID field in order to specify the area feature where each triangle belongs (a containment check algorithm can be applied for this purpose).

(6) In the structure of the SVVM, there is only one arc table; in the UNS we have three, i.e., R4, R8, R9. To create them, first the structure of the original arc table is changed by adding ARCNR according to figure 4. Eventually this table will become R4 (see preceeding section). Next the ARCTRI table is created and the data values are entered of left and right triangles in each record. To automate this updating, an algorithm which computes the turning direction from one arc to the next can be used (see Shmutter and Doytsher, 1988). The ARCLINE table (R9) is created next and the data values of ALID are obtained by transferring those arcs that represent a line feature from the original arc table. The last step is deleting from the arc table those fields which must not be in the table R4.

(7) Extending the structure of the node table by adding a data field for z-coordinate, then updating this data field according to the output from step (2).

(8) Adding two new nodes in the NODE table where the "ROAD2" crosses "RAILROAD" and "RIVER3" using the same planimetric coordinates (x, y) but different height (z-coordinate), and three new arcs as follows:

Existing Nodes				New Nodes				New Arcs		
NODENR	X	Y	Z	NODENR	X	Y	Z	ARCNR	BEG	END
20	32	7	35	40	32	7	40	105	21	40
25	32	12	36	41	32	12	41	104	40	41
								105	41	27

Updating the ARCLINE table by changing the arcs that belong to "ROAD2" (say composing a bridge on "ROAD2") as follows:

Existing arcs		Updated arcs	
ARCNR	ALID	ARCNR	ALID
19	5	103	5
29	5	104	5
38	5	105	5

(9) Dropping the line-crossed table (R6) of SVVM.

The metamorphosis of the SVVM (figure 3) obtained by the above steps is shown in figures 4 and 5.

Query examples

(1) Question: How large is the forest area represented on this map?

Approach: Retrieve all triangles from TRIAREA table which belong to the area class forest (AID = 'FOREST') and then compute the summation of the area of each triangle by first retrieving the three vertices of each triangle.

```
SQL: SELECT trinr
        FROM   triarea, area
        WHERE  aclass = 'FOREST'
        AND    aid = taid;
```

Result: The output from this stage is all the triangle numbers that belong to the forest area.

```
TRIAREA->TRINR
2, 6, 10, 20, 27, 47, 52, 53, 58, 59, 60, 61,62
```

(compare figure with 3 and 5)

The next step is to retrieve coordinates of the vertices of these triangles (TN).

```
SQL: SELECT DISTINCT arcnr
        FROM triarea, arctri
        WHERE (trinr = TN)
        AND ((trinr = ltri) OR (trinr = rtri))
        SAVE TO TEMP triside;
        SELECT DISTINCT nodenr, x, y, z
        FROM arc, node, triside
        WHERE   (triside.arcnr=arc.arcnr)
        AND ((nodenr = beg)
        OR (nodenr = end));
```

Result: If TN is 20, the output is as follows:

NODENR	NODE->X	NODE->Y	NODE->Z
3	8	28	40
4	10	24	35
39	6	12	47

(compare with figure 5)

The area of each triangle can be computed by using various formula; the summation of all triangles that belong to forest area is the area of the forest in this map.

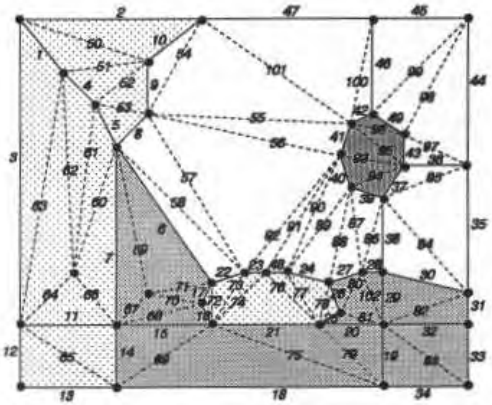


Figure 4 : SVVM after decomposition into triangles showing all arc numbers

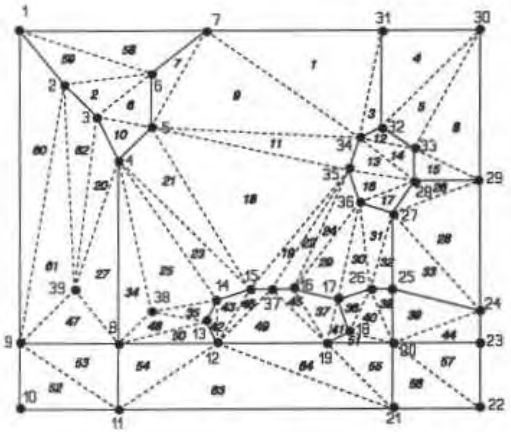


Figure 5 : Triangular Irregular Network showing triangle and node numbers

The first question can be extended to the elevation problem such as:

(2) Question: Give all forest above 40 m.

Approach:

(a) Retrieve all forest areas using the approach as demonstrated in question (1) with the additional criterion that at least one of the three nodes of a triangle must have a z-coordinate greater than 40. The result can be saved in the view or temporary table.

(b) Using the previously saved view or temporary table as an input, separate triangles with all three nodes having z-coordinate greater than or equal to 40 m (save as GT40a

table) from those triangles which have at least one node with a z-value smaller than 40 (save as VERIFY table).

(c) Using VERIFY table as an input, trace all contours at 40 m through this set of triangles and at the same time decompose them into new triangles using the derived contour line as triangle edges. Applying the same procedure as in step (b), separate triangles into two sets: one with all nodes having z-values greater than or equal to 40 (save as GT40b table) and the other one with all nodes having z-values less than or equal to 40 (save as LT40 table).

(d) The union of the two triangle sets obtained from steps (b) and (c), namely GT40a and GT40b respectively, comprises the forest area above 40 meter.

(e) Using the combined data from (d), compute the area of the forest above an elevation of 40 meters.

Note: The SQL example of question (2) is not given because it is almost the same as demonstrated in question (1). Moreover, the contour tracing algorithm needs to be implemented first. Details and programming example can be found in (Alias, 1992).

(3) Question: Give all the triangles which are the neighbours of triangle number 18.

Approach: Retrieve the three vertices of this triangle from TRIAREA table and then match with the begin and end nodes of the arc from the arc table. The left or right triangle of this arc, which has a number different from 18, is one of the neighbours of triangle 18.

```
SQL: SELECT trinr, ltri, rtri
       FROM triarea, arctri
       WHERE ((ltri = 18) OR (rtri = 18))
       AND   (trinr = 18);
```

Result:

```
TRIAREA->TRINR ARC->LTRI ARC->RTRI
        18        11        18
        18        18        21
        18        19        18
```

The above result implies that the neighbours of triangle 18 are 11, 21, and 19 (compare with figure 5).

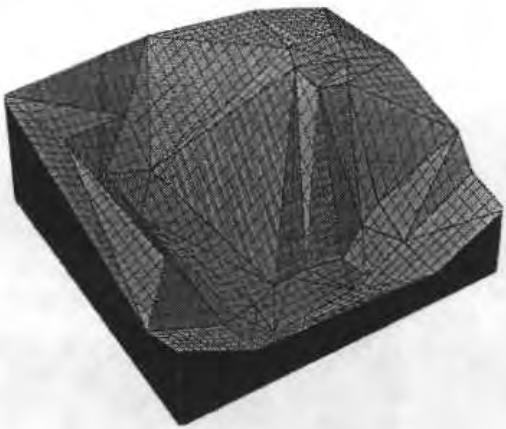


Figure 6 : Perspective view of the terrain.

(4) Question: Which line features are crossed by a specific one?

SQL:

```
(i) SELECT nodenr,x,y,z
FROM node,arc,arcline
WHERE (linid=alid)
AND (arcline.arcnr=arc.arcnr)
AND ((nodenr = beg) OR (nodenr=end))
SAVE TO TEMP nodesofl;
```

```
(ii) SELECT distinct node.nodenr,node.x,node.y,node.z
FROM node,nodesofl
WHERE (node.x=nodesofl.x)
AND (node.y=nodesofl.y)
AND (node.z<>nodesofl.z)
SAVE TO TEMP crosnode;
```

```
(iii) SELECT distinct alid,lclass
FROM arcline,line,arc,crosnode
WHERE ((crosnode.nodenr=beg) or (crosnode.nodenr=end))
AND (arc.arcnr=arcline.arcnr) AND (alid=lid)
AND (alid<>linid);
```

Result: If linid = 5 (ROAD2) the output is

```
ARCLINE->ALID LINE->LCLASS
        3 RAILROAD
        4 RIVER3
```

For more queries see Appendix B.

DISCUSSION

From the above examples, We can easily see that the extended data model maintains the obvious topological characteristics of SVVM. Therefore, most of the queries which are possible on SVVM should also be possible with UNS (see Appendix B). Moreover, it can be seen that the unified of TIN-DTM and GI data structures leads to a more straightforward way of utilizing both types of information, since overlaying of SVVM features with the DTM every time after a query is not needed nor is a large necessary programming effort. In general, UNS shows flexibility and ease of implementation, especially when mapping into a relational database and using a commercial DBMS such as dBASE IV. Furthermore, the two datasets are said to be fully integrated into one, thus data handling and manipulation are simpler.

If there is a requirement for higher fidelity of surface representation than just furnishing the terrain features with z-coordinates, both bulk elevation points and surface-skeleton features have to be added to the database, thus, reducing sampling distance. These additional objects should still be maintained within the TIN structure by densifying the existing triangles. For this purpose, the approach proposed by Jackson (1989) can be applied.

Issues of 3-D topology, such as a road passing over a river, can be handled without the additional table that is needed in SVVM. The problem of line features having different elevation at their crossing is solved by having two nodes, the pair having identical (x, y) coordinates but different z-values. The z-value of the crossing over road is interpolated along the arc of the road at the bridge. Consequently this arc is then split to two new arcs. The bridge node and the two arcs that share it are not part of a triangle. The crossing is found by querying the nodes that have the same (x, y) coordinates. This approach can be extended to also handle "crossing" area features by allowing triangle vertices to carry multiple z-values. However, handling multivalue surfaces, in general, still requires further investigation.

The UNS is expected to have favourable properties in updating of information. Studying this issues requires to establish the consistency rules (see Kufoniyi, 1989). A specific problem arises when a new feature is added as triangles need to be densified. This requires special care in the selection of the interpolation algorithm when a new node is being added; the height information has to be attached to that node; otherwise the integrity of surface representation may be lost. Another challenge is to develop the means for data handling of this new structure to be able to deal with operations like insertion, deletion, etc., which normally alter the topology.

CONCLUSIONS AND OUTLOOK

A data model for the integration of DTMs and GISs has been proposed. Mapping from this data model into a relational data structure is illustrated. The obtained result shows a more complete integration between DTM and GIS than can be achieved by tools for exchanging data between both subsystems. The two datasets are no longer separated. All surface features can be represented with 3-D coordinates, which offers additional possibilities in spatial analysis leading to better utilization of geoinformation. The comparison with SVVM shows that UNS is compatible with SVVM in preserving 2-D topology of a planar map and thus is expected to support the same queries as SVVM. Some discussion about problems relating to multivalue surfaces (crossing but not intersecting features) is also given. Further research is directed to the investigation and development of appropriate data handling and manipulating algorithms with regard to the proposed data model. Another issue to be investigated is to what extent adding redundancy will increase the responsiveness. Under study already is the analysis of the query space of UNS and further extension of the model, such as association with complete 3-D models.

ACKNOWLEDGEMENTS

The authors wish to thank Prof. M. Molenaar for his valuable suggestion specifically concerning the data model, and V. Bric for his contribution to the SQL experiment with the model.

REFERENCES

Alias, 1992. Triangular Network in Digital Terrain Relief Modelling, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Arc/Info, 1991. Surface Modelling with TIN, Arc/Info user's guide, ESRI, U.S.A.

Bouloucos. T, Kufuniyi, O., and Molenaar, M., 1990. A Relational Data Structure for Single Valued Vector Maps, International Archives of Photogrammetry and Remote Sensing, Vol. 28, Part 3/2, Commission III, Wuhan, China, pp. 64-74.

Ebner, H., Hossler, R., and Wurlander, R., 1990. Integration of An Efficient DTM Program Package into Geographical Information Systems, International Archives of Photogrammetry and Remote Sensing, Vol. 28, Part 4, Commission IV, Tsukuba, Japan, pp. 116-121.

Fritsch, D. and Pfannenstein, A., 1992. Conceptual Models for Efficient DTM Integration into GIS, Proceedings Third European Conference on Geographical Information Systems (EGIS '92), Volume. 1, Munich, Germany, pp. 701-710.

Kufoniyi, O., 1989. Editing of Topologically Structured Data, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Jackson, J., 1989. Algorithms for Triangular Irregular Networks Based on Simplicial Complex Theory, ASPRS-ACSM Annual Convention, Baltimore, MD, U.S.A., pp. 131-136.

Mark, D.M., Lauzon, J.P., and Cebrian, J.A., 1989. A Review of Quadtree-based Strategies for Interfacing Coverage Data with Digital Elevation Models in Grid Form, International Journal of Geographical Information Systems, Vol.3, No. 1, Taylor & Francis, London, pp. 3-14.

Molenaar, M., 1988. Single Valued Polygon Maps, International Archives of Photogrammetry and Remote Sensing, Vol. 27, Part B4, Commission IV, Kyoto, Japan, pp. 592-601.

Pilouk, M., 1991. Fidelity Improvement of DTM from Contours, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Pilouk, M., and Tempfli, K., 1992. A Digital Image Processing Approach to Creating DTMs from Digitized Contours, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., U.S.A., pp. 956-961.

Roushannejad, A. A., 1993, Mathematical Morphology in Automatically Deriving Skeleton Lines from Digitized Contours, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Smith, H.C., 1985. Data base design: Composing fully normalized tables from a rigorous dependency diagram, Communication of the ACM, Vol. 28, No. 8, pp.826-838.

Shmutter, B., Doytsher, Y., 1988. An Algorithm to Assemble Polygons, ACSM-ASPRS Annual convention, St. Louis, Missouri, pp. 98-105.

Thiessen, A.H., 1911, Precipitation averages for large areas. Mon. Wea. Rev. July, pp. 1082-1084.

APPENDIX A

POINT TABLE

PID	PCLASS	PAID	PNODE
1	MILL	0	37
2	WELL	2	38
3	CHAPEL	1	39

LINE TABLE

LID	LCLASS
1	RIVER2
2	RIVER1
3	RAILROAD
4	RIVER3
5	ROAD2
6	ROAD3
7	ROAD1

AREA TABLE

AID	ACLASS
1	FOREST
2	GRASSLAND
3	ARABLELAND
4	CITY
5	LAKE
9	OUTERLIMIT

APPENDIX B

Give all arcs bound area "maid".

```
SELECT arcnr
FROM arctri,triarea
WHERE ((ltri=trinr) OR (rtri=trinr))
AND (taid=maid)
GROUP BY arcnr
HAVING COUNT(arcnr)=1;
```

Give all arcs bound map.

```
SELECT distinct arcnr
FROM arctri
WHERE (ltri=0) OR (rtri=0);
```

Which area(s) touch by area "maid" ?

```
SELECT arcnr
FROM arctri,triarea
WHERE ((ltri=trinr) OR (rtri=trinr))
AND (taid=maid)
GROUP BY arcnr
HAVING COUNT(arcnr)=1
SAVE TO TEMP bound;

SELECT distinct taid,aclass
FROM triarea,bound,arctri,area
WHERE (taid<>maid)
AND (arctri.arcnr=bound.arcnr)
AND (taid=aid)
AND ((ltri=trinr) OR (rtri=trinr));
```

Which area(s) bounded by line "linid" ?

```
SELECT distinct taid,aclass
FROM arctri,triarea,arcline,area
WHERE (arcline.alid=linid) AND (taid=aid)
AND (arcline.arcnr=arctri.arcnr)
AND ((ltri=trinr) OR (rtri=trinr));
```

Which area(s) intersected by line "linid" ?

```
SELECT arcline.arcnr,taid
FROM arctri,arcline,triarea
WHERE (arcline.alid=linid)
AND (arcline.arcnr=arctri.arcnr)
AND ((ltri=trinr) OR (rtri=trinr))
SAVE TO TEMP allarcs;

SELECT distinct arcnr,taid
FROM allarcs
GROUP BY arcnr,taid
HAVING COUNT(arcnr)=2
SAVE TO TEMP inter;

SELECT distinct taid,aclass
FROM inter,area
WHERE (taid=aid);
```

Which area(s) intersected by line "linid" ?

```
SELECT arcline.arcnr,taid
FROM arctri,arcline,triarea
WHERE (arcline.alid=linid)
AND (arcline.arcnr=arctri.arcnr)
AND ((ltri=trinr) OR (rtri=trinr))
SAVE TO TEMP allarcs;

SELECT distinct arcnr,taid
FROM allarcs
GROUP BY arcnr,taid
HAVING COUNT(arcnr)=1
SAVE TO TEMP inter;

SELECT distinct taid,aclass
FROM inter,area
WHERE (taid=aid);
```

Is area "aid2" inside of area "aid1" ?

```
SELECT arcnr
FROM arctri,triarea
WHERE ((ltri=trinr) OR (rtri=trinr))
AND (taid=aid1)
GROUP BY arcnr
HAVING COUNT(arcnr)=1
SAVE TO TEMP bound3;

SELECT arcnr
FROM arctri,triarea
WHERE ((ltri=trinr) OR (rtri=trinr))
AND (taid=aid2)
GROUP BY arcnr
HAVING COUNT(arcnr)=1
SAVE TO TEMP bound4;
mnofarc=0
SELECT COUNT(*) into mnofarc FROM bound4;
insert into bound3
SELECT * FROM bound4 ;

SELECT arcnr
FROM bound3 GROUP BY arcnr
HAVING COUNT(arcnr)=2
SAVE TO TEMP commarcs;
mcommon=0
SELECT COUNT(*) into mcommon FROM commarcs;
? " "
IF mcommon=0
 ? "  NO, SORRY !"
ELSE
 IF mcommon=mnofarc
  ? "  YES, BINGO !"
 ELSE
  ? "  NO, SORRY !"
 ENDIF
ENDIF
```

Note: The reader can obtain the results by first implementing the R1 to R9 tables.

CONSEQUENCES OF USING A TOLERANCE PARADIGM IN SPATIAL OVERLAY

David Pullar
Environmental Systems Research Institute
380 New York Street
Redlands. CA 92373
dpullar@esri.com

ABSTRACT

Geometrical algorithms for spatial overlay incorporate a fuzzy tolerance to remove spurious polygons. This fuzzy tolerance may be related to numerical limits for calculating intersections, or related to the variation in the geometry of objects they represent. When the distance between two objects is below the tolerance they are classified as coincident and moved to the same location. Yet using a tolerance as a heuristic to handle decisions of approximate coincidence has some undesirable consequences. Some of the problems encountered are that objects creep outside their allowable tolerance, or the objects geometry is corrupted so that proper topological relations cannot be reconstructed. This paper examines the flaws with applying a tolerance paradigm in spatial overlay, and describes the conditions needed to safely evaluate point coincidence.

INTRODUCTION

Spatial overlay is an analytical tool used to integrate multiple thematic layers into a single composite layer. This involves intersecting all the geometrical objects from each layer and filtering the geometry to remove spurious polygons in the output. This geometrical filtering process relates to techniques for automated scale changing called *epsilon filtering* [Chrisman 1983]. The two geometrical criteria it imposes on the output are; i) creep - no point is moved more than epsilon, and ii) shrouding - no points are left within a distance epsilon. The epsilon distance in map overlay is the geometrical tolerance.

The basis of the filtering process is to resolve point coincidences. That is, if two points are found to be within the geometrical tolerance to one another they are merged and identified as a single point. Most commercial GIS's use a single tolerance as an input parameter to the overlay program for classifying incidence and coincidence relations between objects. Yet there is a need to distinguish between objects that have a well defined geometry with a small tolerance, and objects that have an imprecise geometry. That is we need to distinguish between multiple cases in the same layer where; i) two points are distinct objects separated by a small distance, and ii) two points are meant to represent the same object when they are a small distance apart. This is the rationalization for multiple tolerances.

Most algorithms employ some *tolerance paradigm* to detect coincident points. The tolerance paradigm is a heuristic that says if two quantities are near enough in value then treat them as the same. A simple example will demonstrate the problem with this reasoning paradigm for geometrical algorithms. Figure 1 shows a set of points and their tolerance environments. Each point, called an *epsilon point*, is given by a tuple (x,y,ε) representing an x,y-coordinate and a tolerance radius ε. We arbitrarily chose one epsilon point and begin

testing for point coincident based on the tolerance regions overlapping. When overlapping epsilon points are found they are shifted to coincide exactly. If the points are tested and moved in the sequence shown in figure 1 then it is apparent points will creep from their desired location. One also could imagine the epsilon points as connecting line segments, and the segments degenerating to a point.

The flaw in our reasoning for the tolerance paradigm is in the transitivity of point coincidence. That is, we cannot naively assume that if ($P_1 = P_2$) and ($P_2 = P_3$) implies ($P_1 = P_3$) when the test used to evaluate equality is approximate.

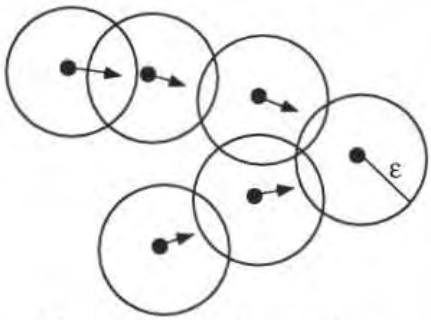


Figure 1. Testing point coincidence

Applying the tolerance paradigm naively causes algorithms to fail or give an incorrect result. And even if applied judiciously the tolerance paradigm has some severe consequences. Recent research in the field of solid modelling has discussed this topic at length. Robust algorithms for intersecting polyhedral objects are known, but it is admitted in a worst case scenario the algorithms will allow objects to collapse to a point [Milenkovic 1989]. This degenerate behavior for computing intersections has even worse consequences when the tolerances become larger and there are multiple polygons intersected. The cause of the problem is in the way points are tested for approximate coincidence before moving them.

CONDITIONS FOR POINT COINCIDENCE

The conditions for resolving coincidences between epsilon points with multiple tolerances is similar to the conditions stated for epsilon filtering. Namely, two geometrical criteria are imposed on the output; i) creep - no point is moved more than epsilon, and ii) shrouding - no points are left within a distance epsilon. The question arises as to the conditions to be fulfilled when dealing with multiple tolerances?

We can define the first criteria creep with respect to multiple tolerances in a straight forward way. A point cannot be moved a distance greater than its epsilon tolerance. What is not straight forward is how to apply the shrouding criteria and make sure points are separated by some minimum distance. This raises two questions when dealing with multiple tolerances;

1. What tolerance will be used to compare two epsilon points?
2. How is the tolerance updated when merging epsilon points?

One obvious way to deal with the first question is to say points are coincident if their tolerance regions overlap. Another way is to say epsilon points maintain a minimum

separation distance. We explore both possibilities.

Overlapping Tolerance Regions

Lets assume the shrouding criteria is based on a geometrical condition, namely the tolerance regions for two points must not overlap. This answers the first question by stipulating that if the sum of the radii for the epsilon regions is greater than the distance between their point centers then they need to be merged. Figure 2 shows this situation.

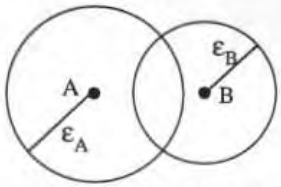


Figure 2. Tolerance environments for two points overlap

To answer the second question we need a way to update the tolerances for two epsilon points. Some different scenarios for updating the tolerance ε_{AB} after merging two points 'A' and 'B' are;

1. $\varepsilon_{AB} = \text{maximum}(\varepsilon_A, \varepsilon_B)$, i.e. the maximum tolerance.
2. $\varepsilon_{AB} = \text{minimum}(\varepsilon_A, \varepsilon_B)$, i.e. the minimum tolerance.
3. $\varepsilon_{AB} = 2/[(\varepsilon_A)\text{-}1+(\varepsilon_B)\text{-}1]$, i.e. the weighted sum of the two tolerances.
4. $\varepsilon_{AB} = \varepsilon_A \cup \varepsilon_B$, i.e. the smallest enclosing sphere within their union.
5. $\varepsilon_{AB} = \varepsilon_A \cap \varepsilon_B$, i.e. the smallest enclosing sphere within their intersection.

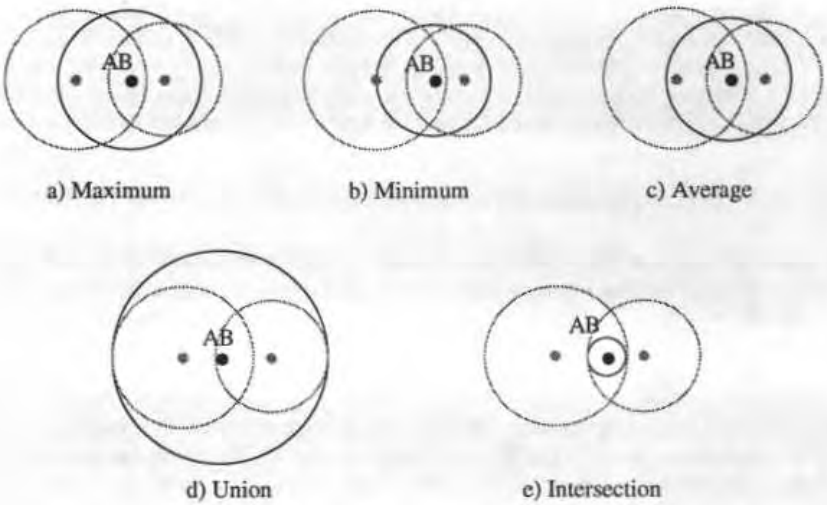
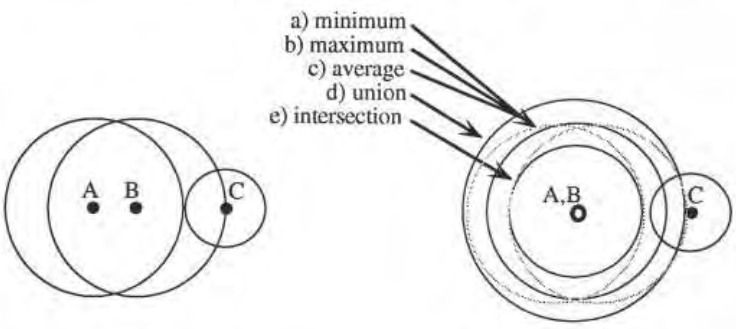


Figure 3. Updating the tolerance regions for point coincidence

Figure 3 shows each of the above methods. By examining some specific point configurations we can easily show that none of the methods are adequate. For instance, consider the three epsilon points 'A', 'B', and 'C' and their associated tolerance environments in figure 4a. Points 'A' and 'B' are found to be close and are merged to form

'AB'. The first four methods of updating the tolerances for 'A' and 'B' - e.g. maximum, minimum, average, and union - all cause overlap with the tolerance region for 'C'. This is illustrated in figure 4b. But if 'C' is merged with 'AB' this will contradict the creep criteria. That is the new point will likely be outside the tolerance region for 'C'. The fifth method, e.g. intersection, is also inappropriate because when the tolerance regions for two points are just overlapping then the updated tolerance will converge to zero. This would easily lead to instability in determining the coincidence between points.



a) Comparing points 'A','B', and 'C' b) Point 'A' and 'B' are merged

Figure 4. Coincidence Test

Therefore our first approach for defining the shrouding criteria has some basic flaws. None the less, some researchers in the field of computational geometry have used approaches similar to this in their work. Segal [1990] uses the union of tolerance regions to analyse point coincidence, but does so at the expense of relaxing the creep criteria. Fang and Bruderlin [1991] use a combination of computing both the union and intersection of the tolerance regions for detecting ambiguities when analysing point coincidences. The algorithm is re-run when an ambiguity is discovered using different estimates for the tolerances.

Minimum Separation

From simple reasoning we have shown the flaw in merging points with overlapping tolerance regions to enforce the shrouding criteria. An alternative to is base the shrouding criteria on maintaining a threshold separation between two points. The separation distance must be related to the tolerances of the points. An obvious possibility is to use any one of the first three criteria for updating clusters discussed in the last section, namely the minimum, maximum or average tolerance for the points. Therefore an alternative test for shrouding is that the separation d_{AB} between two points 'A' and 'B' be less than;

1. $d_{AB} < \text{maximum}(\varepsilon_A,\varepsilon_B)$, i.e. the maximum of the two tolerances.
2. $d_{AB} < \text{minimum}(\varepsilon_A,\varepsilon_B)$, i.e. the minimum of the two tolerances.
3. $d_{AB} < 2/[(\varepsilon_A)\text{-}1+(\varepsilon_B)\text{-}1]$, i.e. the weighted average of the two tolerances.

It is natural to assume the updated tolerance for an output point is also computed from the respective maximum, minimum, or average of the tolerances. One could easily expect the first possibility of using the maximum of the two tolerances is violated by inspection of figure 4. Less obvious is the fact that using a weighted average as a shrouding criteria and then updating the tolerances also will violate the creep criteria for certain input data. We do

not prove this, but have found this to be the case from running several tests with randomly generated data sets. Rather, we will set out to prove that a lower bound using the minimum tolerance as a separation criteria may be achieved. From empirical testing we could find no test case that violated this condition when merging epsilon points in the most compact way.

Thus, this paper proposes that coincidence of epsilon points may be solved in a consistent way to satisfy the following geometrical criteria;

1. An input point cannot be moved more than its epsilon tolerance to merge with an an output point, i.e. if 'A' is moved to 'B' then the distance $d(A,B) < \varepsilon_A$.
2. Epsilon points in the output set are separated by a minimum distance, this lower bound is determined by pairwise comparisons to be the minimum epsilon tolerance, i.e. output epsilon points 'A' and 'B' are separated by at least the distance $\text{minimum}(\varepsilon_A,\varepsilon_B)$.
3. When two epsilon points are merged a new point center is located at their mean position, and a new tolerance is computed as the minimum of the two epsilon tolerances, i.e. if epsilon points 'A' and 'B' are merged then $\varepsilon_{AB}=\text{minimum}(\varepsilon_A,\varepsilon_B)$.

We have found that these are the only feasible conditions that may be met for unambiguously solving point coincidence. To show this, we first re-state the point coincidence problem in a more formal way. We use concepts from graph theory to solve for point coincidence as a graph location problem. We then prove that the three conditions stated above are satisfied.

A MODEL FOR POINT COINCIDENCE

To prove the minimum separation criteria is valid for all inputs we need to define the properties of output points that satisfy the point coincidence relations. The best way to describe this problem, and its solution, is as a point clustering problem. Hartigan [1975] describes clustering as the 'grouping of similar objects'. In our case the objects are points with their associated tolerance, and they are grouped to new centers that satisfy the coincidence relations. The clustering is also chosen to maximize the separation criteria by stipulating that point coincidences minimize some measure of dissimilarity. Hartigan describes several dissimilarity measures, one popular method is to group the elements in a way that minimizes the spread of points in each group. Minimizing the spread is interpreted as minimizing the sum of the squared lengths from cluster points to their center. This is called a *sum-of-squared-error clustering* [Duda and Hart 1973].

The P-median Location Problem

The sum-of-squared-error clustering is closely related to finding the medians of a graph, this is called the *p-median* problem [Christofides 1975]. The Euclidean p-median problem is described in the following way. Given a set $X=\{p_1, p_2, ..., p_n\}$ of n points (x,y), find a set X' of m points $\{p'_1, p'_2, ..., p'_m\}$ so as to *minimize* the expression;

$$\sum_{i=1}^{n} \min_{1 \le r \le m} \{ \| p'_r - p_i \| \} \qquad (1)$$

where ||..|| designates the metric distance between point centers.

Intuitively, we wish to minimize the sum of the radii that enclose points of X by circles located at centers of X'. We also refer to the points of X' as cluster centers.

The p-median problem has some nice set-theoretic implications. The set of points from X associated with a cluster center define a *set-covering* of X. The points of X are grouped into sets $X_1,...,X_m$ according to way the circles located at cluster centers of X´ cover points in X. These are called the *covering sets*, such that;

$$\bigcup_{r=1}^{m} X_r = X \qquad (2)$$

In addition the sets $X_1,...,X_m$ are pair-wise disjoint. This is called a *set-partitioning* of X, such that;

$$X_i \cap X_j = \varnothing, \quad \forall i,j \in \{1,...,m\} \qquad (3)$$

These set-theoretic properties are used to define point coincidences in a consistent way. We will adapt the p-median problem to deal with epsilon points. We now re-state the problem, and call it a *distance constrained* p-median problem.

Constrained Clustering

The distance constrained Euclidean p-median problem is described in the following way. Given a set $X=\{p_1, p_2,..., p_n\}$ of *n* epsilon points (x,y,ε), find a set X´ of epsilon points $\{p'_1, p'_2,..., p'_m\}$ where $m \leq n$ so as to *minimize* the expression;

$$\sum_{i=1}^{n} \min_{1 \leq r \leq m} \{ \| p'_r - p_i \| \}, \quad \| p'_r - p_i \| < \varepsilon_i \qquad (4a)$$

and

$$\| p'_r - p'_s \| < \text{minimum } (\varepsilon'_r, \varepsilon'_s), \quad \forall r,s \in \{1,...,m\} \qquad (4b)$$

where ||..|| designates the metric distance between epsilon point centers.

As in (1) we are minimizing the sum of the radii that enclose points of X by circles located at cluster centers of X´. But equation (4a) requires the distance between a cluster center p'_r and an input point p_i is constrained to be less than the epsilon tolerance ε_i for that point. Equation (4b) additionally stipulates that a minimum separation is maintained between clusters centers.

Notice that a variable number of cluster centers $m \leq n$ is permitted. The problem now resembles a clustering procedure rather than a graph location problem, for this reason we refer to the solution of the distance constrained Euclidean p-median problem simply as a *constrained clustering*. The lower limit to the number of cluster points *m* is determined by the minimum number of points that will define a set-covering of X based upon equation (4b). The number of clusters is given by the solution set that satisfies both conditions.

The constrained clustering defines a set-partitioning of X. Each epsilon point of X is clustered to the nearest cluster point from the set X´. So any two points p_i,p_j clustered together are considered part of the same equivalence class based on the relation p_i *is-coincident-to* p_j. Since coincidence is an equivalence relation then by definition the relation is reflexive, symmetric and transitive. This property is used to avoid the inconsistencies

found in naive algorithms for comparing points.

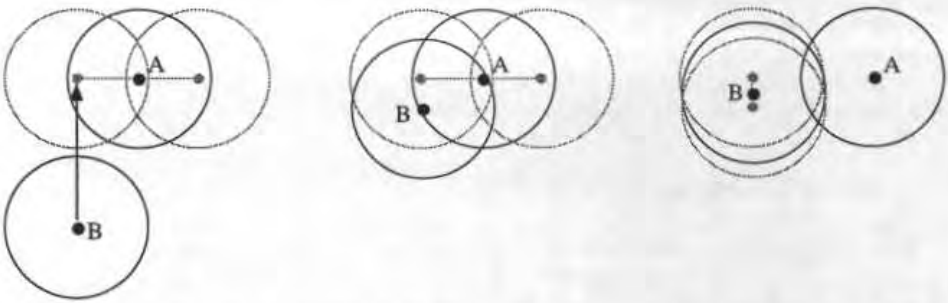
Proof of Constrained Clustering Conditions

We need to prove a constrained clustering fulfills the three geometrical conditions stated in the last section. These conditions will be defined in terms of a clustering procedure to decide point coincidence relations on a set of epsilon points.

Definition. Given a set of epsilon points $X=\{p_1, p_2, ..., p_n\}$ we partition X into subsets $X_1,...,X_m$. Each subset X_k defines a cluster set with a representative point p'_k chosen as the cluster center, the set of *m* cluster centers is denoted as the set X´. The relation between the epsilon points in the sets X and X´ is called a constrained clustering when the following conditions are satisfied;

1. $\| p'_r - p_i \| < \varepsilon_i$ (from eq. 4a)
2. $\sum_{i=1}^{n} \min_{1 \le r \le m} \{ \| p'_r - p_i \| \}, \quad \| p'_r - p_i \| < \varepsilon_i$ (from eq. 4a)
3. $\| p'_r - p'_s \| <$ minimum $(\varepsilon'_r, \varepsilon'_s), \quad \forall r,s \in \{1,...,m\}$ (from eq. 4b)

Proof. Lets assume a constrained cluster exists satisfying conditions 1 and 2. For now lets disregard condition 3, and allow clusters to be separated by less than the minimum tolerance. Without loss of generality, assume there are two cluster centers $p'_r, p'_s \in X'$ which are separated by a distance less than minimum$(\varepsilon'_r, \varepsilon'_s)$. There must also exist extreme[1] points belonging to the cluster sets $p_i \in X_r$, $p_j \in X_s$ which lie within the distance minimum$(\varepsilon'_r, \varepsilon'_s)$ to one another. Since these points are within the minimum tolerance to one another they are free to group together without violating condition 1, and form a new cluster with a smaller diameter. Therefore, there must be another way of clustering the points which gives a smaller sum of lengths between cluster centers and points from X. This violates condition 2, and to avoid the contradiction we conclude condition 3 must be true.



a) Move 'B' closer to 'A' b) 'A' and 'B' within minimum tolerance c) Re-clustering to satisfy minimum diameter clusters

Figure 5. Demonstrates proof of constrained clustering

As an example we can examine the arrangement of points shown in figure 5. Diagram a) shows two clusters that satisfy conditions 1-3. We examine the effect of moving the two

[1]Extreme points are the points on the convex hull of a set of points [Preparata and Shamos 1985]

clusters closer together, until in diagram b) they are separated by less than their minimum tolerances. It is evident that the points could be re-clustered to obtain a more compact cluster with a smaller sum of lengths between cluster centers and input points, as shown in diagram c).

This means we can place an upper bound on the distance points can be moved, and a lower bound on the separation between final point centers. Even more importantly, we have defined the coincidence relation based upon the constrained clustering. Points that belong to the same cluster $\{p_i,p_j \in X_r\}$ are coincident, otherwise they are non-coincident. By using this set membership relation to test for point coincidence we satisfy the transitivity rules and avoid inconsistencies.

CONCLUSION

This paper has examined how a tolerance paradigm is used in geometrical algorithms. Each point is assigned a tolerance and is referred to as an epsilon point. The tolerance paradigm determines what epsilon points are coincident, and subsequently merges them. In determining coincidences certain geometrical properties are desired; these are i) creep to prevent input points from drifting too far, and ii) shrouding to maintain some separation between points. The paper focuses on a way to test for coincidence of epsilon points with multiple epsilon tolerances, and several solutions are discussed. We found one solution that uses a clustering approach to merge epsilon points is a suitable model for point coincidence.

The problem of clustering epsilon points, which we named constrained clustering, is described using set-theoretic principles from graph location theory. We describe a variation of the Euclidean p-median problem which constrains the distances between points to satisfy the two geometrical criteria for creep and shrouding. Using this approach, point coincidence is defined as an equivalence relation over a set of epsilon points. This allows us to make set-theoretic conclusions about epsilon points. For instance, we can now unambiguously say that if (P_1 is-coincident-to P_2) and (P_2 is-coincident-to P_3) implies (P_1 is-coincident-to P_3).

Being able to cluster epsilon points and guarantee geometrical properties has beneficial consequences for designing a multi-tolerance overlay algorithm. It provides verification conditions that is used in a correctness proof for the map overlay algorithm [Pullar 1991].

Another implication of these results is that to obtain upper and lower bounds for creep and shrouding we must solve a geometric location problem. The Euclidean p-median problem has a complexity classed as NP-hard, and even an approximate solution requires an efficient clustering algorithm [Megiddo and Supowit 1984].

REFERENCES

Chrisman N., 1983, Epsilon Filtering: A Technique for Automated Scale Change. *Proceedings 43rd Annual Meeting of ACSM*: p.322-331

Christofides N., 1975, *Graph Theory: An Algorithmic Approach.* Academic Press.

Duda R., and Hart P., 1973, *Pattern Classification and Scene Analysis.* Wiley Interscience.

Fang S. and Bruderlin B., 1991, Robustness in Geometric Modeling - Tolerance-Based Methods. *Proceeding International Workshop on Computational Geometry CG'91*, Switzerland, Lecture Notes in Computer Science, #553, Springer-Verlag, Editors H.Beeri and H.Noltemeier. p.85-102

Hartigan J.A., 1975, *Clustering Algorithms*. Wiley, New York.

Megiddo N., and Supowit K., 1984, On The Complexity Of Some Common Geometric Location Problems. *SIAM Journal of Computing* 13(1): p.182-196

Milenkovic V.J., 1989, Verifiable Implementations Of Geometric Algorithms Using Finite Precision Arithmetic. In: *Geometrical Reasoning*, editors D. Kapur and J. Mundy, MIT Press, Pennsylvania.

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

Pullar D.V., 1991, Spatial Overlay with Inexact Numerical Data, *Proceedings Auto-Carto 10*, Baltimore. p.313-329

Segal M., 1990, Using Tolerances to Guarantee Valid Polyhedral Modeling Results. *Proceedings SIGRAPH*: p.105-114

RASTER-TO-VECTOR CONVERSION: A TREND LINE INTERSECTION APPROACH TO JUNCTION ENHANCEMENT

Peng Gao
Michael M. Minami

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

ABSTRACT

One of the problems with automatic or semi-automatic raster-to-vector conversion is dimpled junctions, which are caused by the underlying thinning algorithms. While useful for shape reconstruction from skeleton, dimpled junctions are undesirable for data conversion. Techniques, referred to as junction enhancement, have been developed to deal with, among others, the dimpled junction problem. However, existing junction enhancement techniques lack well-defined procedures and are only able to handle junctions on a case by case basis. This paper presents a trend line intersection approach to systematically enhance all types of junctions. This approach has a set of simple and well-defined procedures for estimating the trend lines of all arcs coming to a junction, and for calculating the new junction location as the intersection of all the estimated trend lines. It can handle junctions with any number of incoming arcs and any type of configuration using the single set of procedures. The experiments have shown that the new approach is very effective in dealing with dimpled junctions.

INTRODUCTION

Scan digitizing is becoming a practical alternative to table digitizing due to the wide availability of scanners (Becker 1992). Unlike table digitizing, however, scan digitizing requires the additional step of raster-to-vector conversion after a map is scanned in the raster form. There are still many unsolved problems with automatic or semi-automatic raster-to-vector conversion. One of them is dimpled intersections, or junctions, caused by the underlying thinning algorithms, as shown in Figure 1 (a). While useful for shape

Figure 1. (a) Dimpled junction; (b) Enhanced junction

reconstruction from skeleton, dimpled junctions are undesirable for data conversion, where a clean intersection is more satisfactory (Figure 1 (b)). Techniques, referred to as junction enhancement, have been developed to deal with, among others, the dimpled junction problem. Most current techniques start by grouping junctions into junction classes, such as, +-junction, T-junctions, X-junctions, Y-junction, etc. A template is pre-defined for each

junction class, and used to match an input junction. If a match is found with one of the pre-defined classes, the input junction is enhanced with the rules associated with that junction class. However, the set of rules for each junction class is pre-defined, and if a junction deviates from its standard form, it will be either incorrectly adjusted or left dimpled. The number of junction classes is also pre-defined, and, if an input junction does not match one of the pre-defined classes, it will be left dimpled, too. It is clear that this approach lacks a systematic procedure for dealing with a specific junction, and lacks flexibility in handling different types of junctions. This paper presents a trend line intersection approach for dealing with the dimpled junctions. Given an input junction containing a node and arcs coming to it, this approach starts by estimating the trend line for each arc within a range close to the node and then, moves the junction to the intersection of trend lines of all the arcs and reconnects the arcs to the new node. This approach has a set of simple, well-defined procedures for estimating the trend line of an incoming arc, and for calculating the intersection of all the estimated trend lines. It can handle junctions with any number of incoming arcs and any type of configuration using the single set of procedures.

JUNCTION ENHANCEMENT

A junction is a structure consisting of a node, a set of arcs, and a relationship between the node and arcs. The location of a junction is the location of the node. Junction enhancement is a process of replacing the old node with a new one and reconnecting arcs to the new node. The trend line intersection approach is based on a simple observation and assumption that the true junction location is the intersection of the trend lines of all arcs coming into a junction. In this section, three important parameters, *range*, *angle tolerance*, and *distance tolerance*, and two concepts, trend line and extended trend line, are defined first, and followed by the description of the procedure for performing the junction enhancement by the trend line intersection.

Range and Trend Line

The *range* defines the segment of an arc within which the trend line is estimated. It is the straight line distance measured from the node to a point on the arc, see Figure 2. The

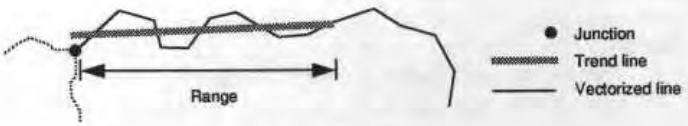


Figure 2. Range and Trend Line

trend line is a line in the form: $y = ax + b$, fitted to the arc within the range. The trend line

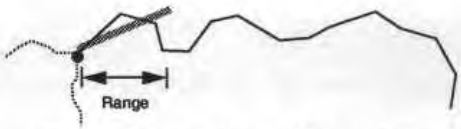


Figure 3. Smaller Range Value and Trend Line

represents the general direction of an arc within the range. The range value controls the sensitivity of the trend line to the local direction around the node. The smaller the range, the more sensitive the trend line will be to the local direction (Figure 3).

Angle Tolerance and Extended Trend Line

The *angle tolerance* specifies the maximum angle, α, between two trend lines such that they can be considered parallel and thus merged into an extended trend line (Figure 4). An *extended trend line* is a single line fitted to the two parallel arc segments from both sides. Figure 4 (b) shows the extended trend line which replaces the two parallel trend lines in Figure 4 (a).

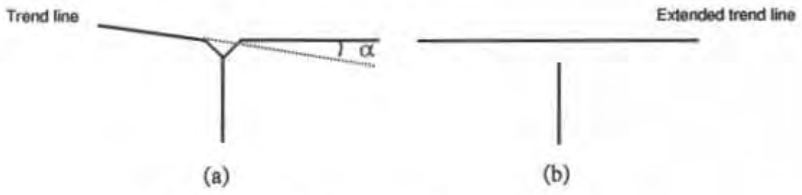


Figure 4. Angle Tolerance and Extended Trend Line

Distance Tolerance and Node

The *distance tolerance* specifies the maximum distance a node can move during junction enhancing (Figure 5). A junction is enhanced only when the distance between the

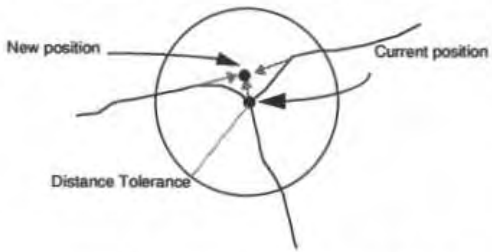


Figure 5. Distance Tolerance and Node

new node position and its current position is within the distance tolerance.

Junction Enhancement Procedure

In general, this procedure calculates a new node location and reconnects all related arcs to the new node to form a clean junction. The new node location is determined by first evaluating the trend of the vectorized lines, or arcs, coming into the junction. A trend line represents the general, or average, direction of the vectorized line based on a subset of the vertices within the line. The number of vertices used to determine the trend line is controlled by the *range* (Figure 2 and 3). Then, all trend lines are checked to determine if any two are parallel and thus should be merged. Two lines are assumed to be parallel if the angle between the trend lines falls within the specified *angle tolerance* (Figure 4). Finally,

the remaining trend lines are intersected to create the new node, see Figure 6, and the

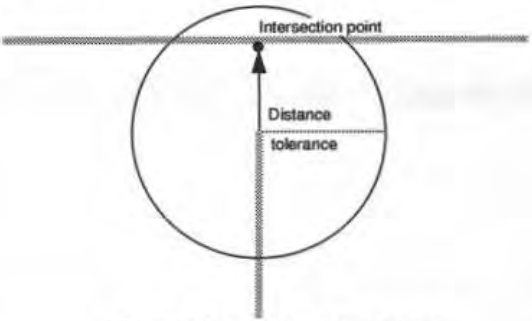


Figure 6. Trend Line Intersection

junction is enhanced if the distance between the current and new nodes is within the *distance tolerance*, as shown in Figure 5.

Assume that there exists a junction structure which contains a node, arcs and the topological relationship between the node and arcs connecting to it. The following is a more formal procedure for the trend line intersection approach.

1) *for each arc coming into a junction, sample the segment of the arc within the range, and fit a line:* $\mathbf{y = ax + b}$*; to the sample points along the arc;*
2) *find all parallel trend lines whose angles are within the angle tolerance, refit an extended trend line for each parallel pair, and replace the two parallel ones with the extended trend line;*
3) *intersect all remaining trend lines (including the extended ones) to determine the node location:*
   a) *if number of extended trend lines is greater than 1, calculate the intersection points between all pairs of extended trend lines;*
   b) *if number of extended trend lines is 1, calculate the intersection points between the extended trend line and the rest of trend lines;*
   c) *if the number of extended trend lines is 0, then calculate the intersection points between the best fitted trend line and the rest of trend lines, where the best fitted trend line is selected based on a goodness of fit value;*
4) *calculate the new node location as the average of those intersection points that are within the distance tolerance to the old node location;*
5) *reconstruct the junction using the new node;*
6) *select another junction and repeat steps 1-5 or stop when there are no more junctions to process.*

It is easy to see that this procedure can be applied to junctions with any number of incoming arcs and with any kind of configuration. Given sample points along an arc, the trend line can be fitted using the Least Squares technique, and the χ^2 value can be used as the goodness of fit value. In general, the extended trend lines yield a more reliable new node location. Therefore, when there is more than one extended trend line, the new node location is simply determined by the intersection of the extended lines. This is the case of +-junctions. When there is only one extended trend line, the extended trend line is used as the base and the rest of the trend lines intersect with it. This is the case of T-junctions and K-

junctions. When there is no extended trend line, the best fitted trend line is used as the base and the rest of the trend lines intersect with it. The new node location created in the absence of an extended trend line is less reliable. This is the case of Y-junctions.

RESULTS

The trend line intersection approach has been implemented and tested in a production system. This section illustrates the results of this approach when applied to artificially constructed junctions, and real junctions from a scanned soil/vegetation composite map. Figure 7 shows four different types of junctions, including +-junction, X-junctions, T-junctions, and Y-junctions, before enhancement., White lines are converted

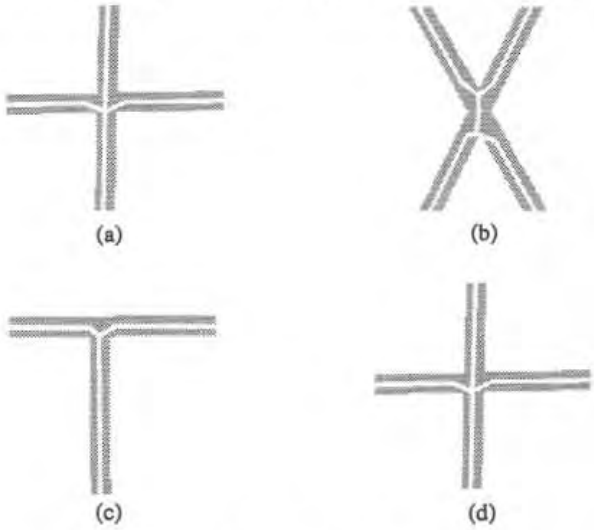


Figure 7. Artificial Junctions before Enhancement

vector lines displayed on top of raster lines in grey. Figure 8 shows the same junctions after enhancement. Figures 9 and 10 contain a small section of a scanned soil/vegetation composite map. Figure 9 shows vectorized junctions and arcs before enhancement, and Figure 10 shows the results after enhancement. Note that there are many new types of junctions in this map section, including K-junctions and 6-valent junctions (a junction with 6 incoming arcs), in addition to different forms of X-, T-, and Y-junctions.

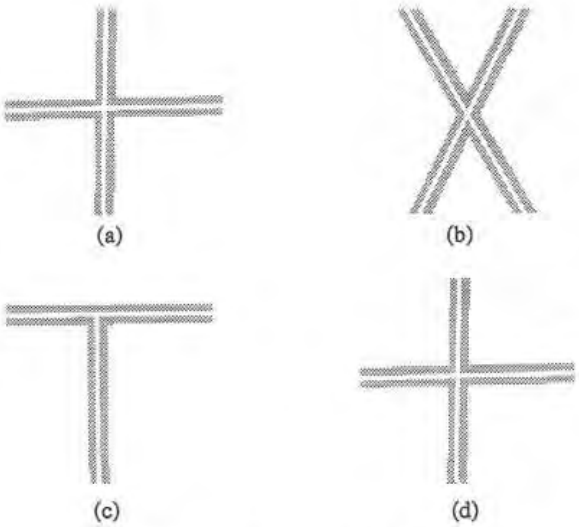


Figure 8. Artificial Junctions after Enhancement

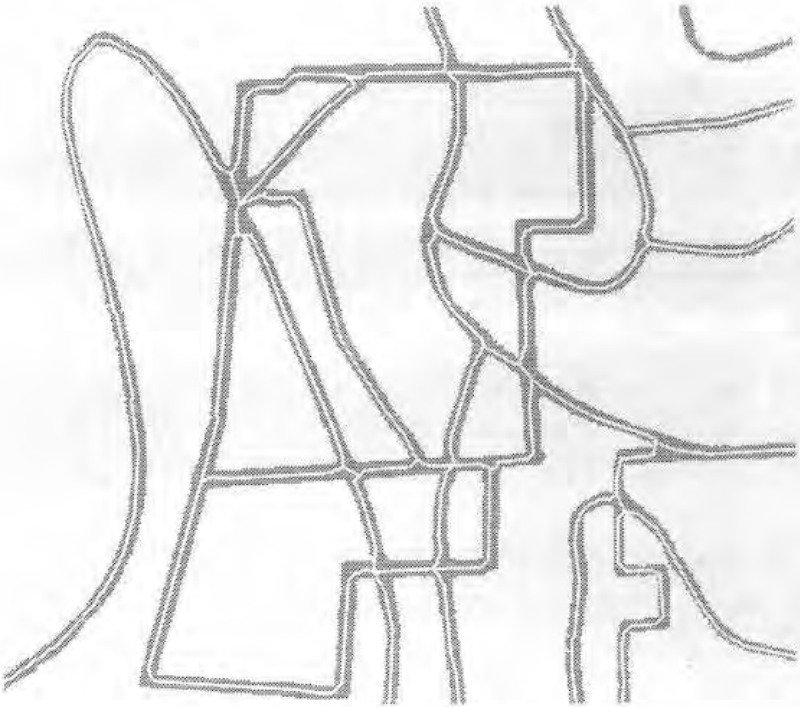


Figure 9. Scanned Soil Map before Enhancement

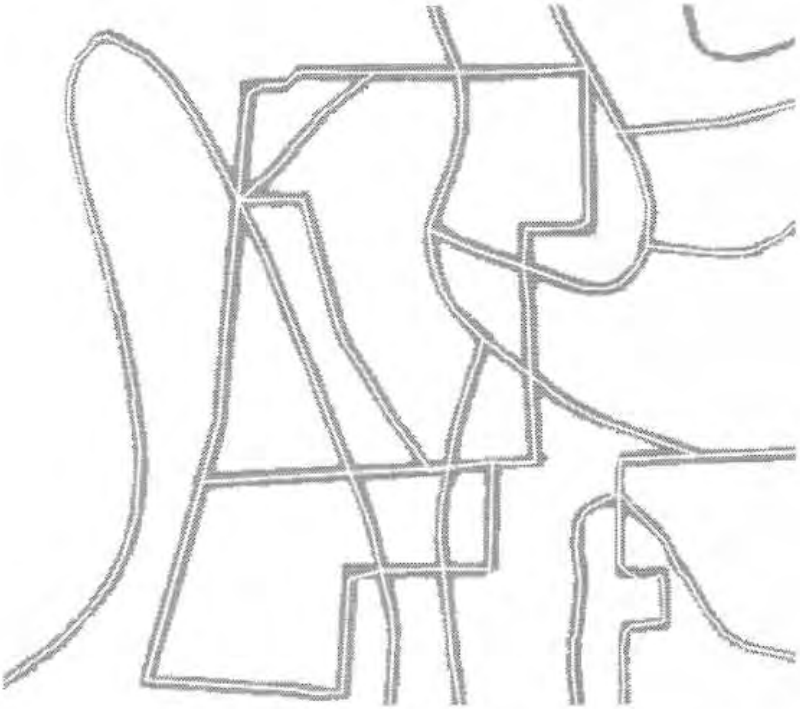


Figure 10. Scanned Soil Map after Enhancement

CONCLUSION

The trend line intersection approach provides a simple, yet effective, procedure for enhancing dimpled junctions, and is applicable to junctions with any number of incoming arcs and with any kind of local configuration. It eliminates the needs for ad hoc rules for handling different types of junctions.

REFERENCES

Becker, R.T, and Devine, H.A. 1992, Raster-To-Vector Conversion of Scanned Document fro Resource Management, *ASPRS/ACSM/RT 92 Technical Papers*, Washington, D.C.

Vector vs. Raster-based Algorithms for Cross Country Movement Planning

Joost van Bemmelen, Wilko Quak,
Marcel van Hekken, and Peter van Oosterom

TNO Physics and Electronics Laboratory,
P.O. Box 96864, 2509 JG The Hague, The Netherlands.
Phone: +31 70 3264221, Fax: +31 70 3280961,
Email: oosterom@fel.tno.nl

Abstract

The problem of cross country movement planning is to find the least cost path between a source and a destination given a polygonal partitioning of the plane. This paper presents the comparison between a vector-based algorithm and several raster-based algorithms for solving the cross country movement planning problem. Besides three known raster methods (standard, more-connected, and quadtree), a new method, *extended raster* approach, has been designed and implemented. The latter finds better solutions because of the higher degree of freedom when moving through a rasterized terrain, that is, more directions can be followed. The vector method of Mitchell and Papadimitriou has also been implemented for comparison. Assuming a reasonable sized terrain with a reasonable accuracy all methods have performance problems. Therefore, a *hierarchical approach* is presented which can be applied in both the vector and raster domain. This approach will find the optimal path in contrast to other hierarchical approaches, which do not guarantee to find the optimal path, but often return a good path. The performance improvement should make the hierarchical approach suitable for interactive movement planning tasks.

1 Introduction

The cross country movement (CCM) problem is also known as the Weighted Region (Least Cost Path) Problem [15, 22]. The unit cost of traversing a given region (polygon) is uniform within the region, but varies between regions and is based on soil, vegetation, and so on. Finding an optimal path, that is, locating a corridor from a given source location to a given destination location [3, 6, 10] is not only used for travelling. It can also be applied in other planning situations, such as building highways, railways, pipelines and other transport systems. The cost function is based on optimization criteria such as time, safety, fuel usage, impact, length, etc. Note that the CCM-problem is very different from the more common linear route planning in a road network, because the cost of using an element (polygon) is not fixed.

Section 2 contains the description of four different raster-based algorithms for the CCM planning problem. The vector-based approach of Mitchell and Papadimitriou

[15] follows in Section 3. Section 4 contains details with respect to the implementation, some test results, and a comparison of the different methods. All presented solutions require a lot of time and space when using realistic data sets. Therefore, an exact solution hierarchical approach is given in Section 5. Finally, conclusions and future work are described in the last section.

2 Raster CCM Algorithms

The first type of solution to the CCM-problem begins with superimposing a regular rectangular grid on the plane [1]. The sampling theories of Shannon, Whittaker and Nyquist [11, 17] state that the gridsize should be at least $2\sqrt{2}$ times smaller than the smallest detail to be kept. After the rasterization a suitability score (weight or cost value) is assigned to each grid-cell. This value represents the "cost per unit distance" travelling inside a cell. Several techniques, such as the Delphi Method and the Nominal Group Technique [9], have been developed to assign the scores. These scores are determined by several geographic variables such as soil type, obstacles, vegetation and overgrowth. Other factors that influence the score are weather conditions and means of transportation: foot, bike, car, tank, etc.

Several raster-based algorithms for finding the shortest path[2] have been developed. All algorithms are based on making a graph with nodes (e.g. raster cells) and implicit edges, the possible movements between the nodes. Then the Dijkstra [1, 4] graph search algorithm is used to find the optimal path. Other search techniques could be used instead, e.g. based on heuristics: A*-algorithm [8]. The main difference between the algorithms is the assignment of nodes and edges:

- standard 4-connected rasters: only rook's moves on a chess board are allowed; see Subsection 2.1;
- more-connected rasters [3, 6, 10]: 8-connected rasters (queen's move are also allowed), 16-connected rasters (queen's+knight's moves), 32-, 64- and 128-connected; see Subsection 2.2;
- extended raster: this new method extends the number of possible angles to move from one individual grid cell to one of its 8-connected neighbors; see Subsection 2.3;
- quadtree [20] based rasters in order to save memory space; see Subsection 2.4.

2.1 Standard Raster Distortions

A number of geometric distortions can be distinguished in the paths produced on a standard raster grid: *elongation*, *deviation* and *proximity* distortion [6, 10]. In Fig. 1 a raster of a uniform terrain, i.e. each cell having the same suitability score, is visualized. A route is planned from raster cell s to raster cell t. The path which is as close as possible to the continuous-space path (see Fig. 3), is a stair-stepped route. The *elongation* error ϵ is defined as the ratio of the cost of the calculated shortest path to the cost of the continuous-space path; e.g. in Fig. 1, $\epsilon = 2/\sqrt{2} = \sqrt{2}$, that is, the calculated shortest path on a 4-connected raster is approximately 41 per cent longer than the continuous-space path.

[1] An alternative grid is a honeycomb grid in which each hexagon has six neighbors [12].

[2] The term "shortest path" has the same meaning as "optimal path" in this paper.

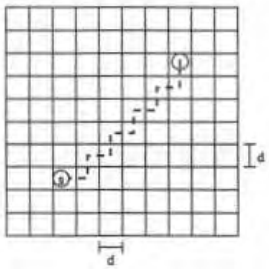


Fig. 1: Stairs effect.

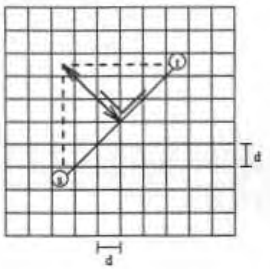


Fig. 2: Max deviation.

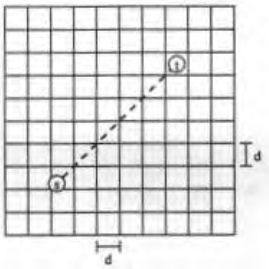


Fig. 3: Real distance.

In general, on a uniform grid, the shortest path will make moves in at most two directions. The maximum *deviation* error δ occurs when all moves in one direction are executed first; e.g. in Fig. 2, $\delta = 0.5$ times the continuous-space path. Note that paths in Fig. 1 and 2 have the same elongation error.

Proximity distortion occurs from the fact that suitability scores are usually calculated for each cell independently of its neighbors. In this way paths can be found which are calculated optimal, but the influence of their neighborhood is ignored. Therefore, paths trough a cheap narrow strip will get the same cost value as paths trough a large uniform terrain. The problem of proximity distortion can be solved by performing a smoothing operation on the suitability score matrix. Note that only the score matrix changes and that the actual problem of route planning does not change.

2.2 More-Connected Rasters

An approach to reduce the distortions in the standard raster is to extend the number of directions a route can follow from each cell. If new implicit edges (queen's moves) are also inserted between every two already existing edges, then a *8-connected* raster is created; see Fig. 4. This process can be repeated to obtain a *16-connected* raster by adding the knight's moves (Fig. 5) and repeated again to obtain a *32-connected* raster; see Fig. 6. In the same manner *64- and 128-connected* rasters can be created.

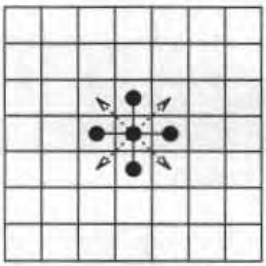


Fig. 4: 8-connected.

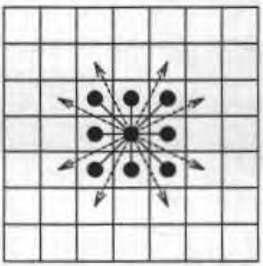


Fig. 5: 16-connected.

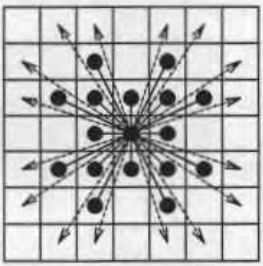


Fig. 6: 32-connected.

In a uniform terrain, the increment in directional possibilities leads to better shortest paths, not only intuitively (Fig. 7, 8, and 9), but also mathematically [10]; see Fig. 10. Fig. 11 visualizes the difference in shortest paths found with different connected raster approaches in a simple terrain. The more-connected raster approach has also several drawbacks. The computation of the cost of an edge is more complicated, because

an edge may pass several cells with line segments with different lengths [24]. Furthermore, more-connected rasters imply more dense graphs, which result in longer computation times.

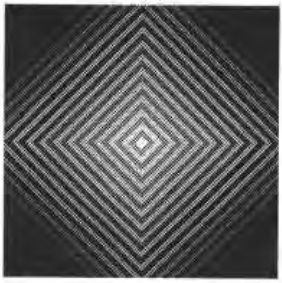


Fig. 7: 4-connected.

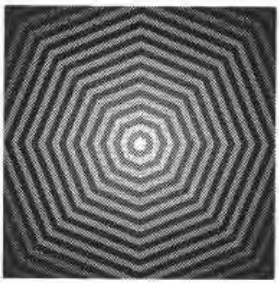


Fig. 8: 8-connected.

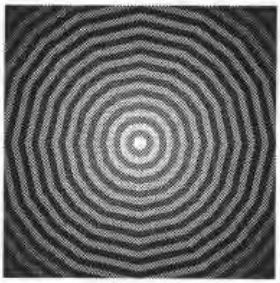


Fig. 9: 16-connected.

Another drawback is that in a non-uniform terrain certain desired angles may be expensive, because the longer edges may also pass very expensive cells. Perhaps the most surprising drawback of the 16- and more-connected raster cell approach is that intersecting paths are possible. One expects that when two shortest paths meet that they will converge and continue as one path to the source. In Fig. 12.a an example of two intersecting paths is given. In this figure the dark colored raster cells have a higher suitability score than the light colored raster cells.

connect	Maximum Elongation ϵ	Maximum Deviation δ
4	1.41421	0.50000
8	1.08239	0.20710
16	1.02749	0.11803
32	1.01308	0.08114
64	1.00755	0.06155
128	1.00489	0.04951

Fig. 10: Maximum elongation and maximum deviation errors.

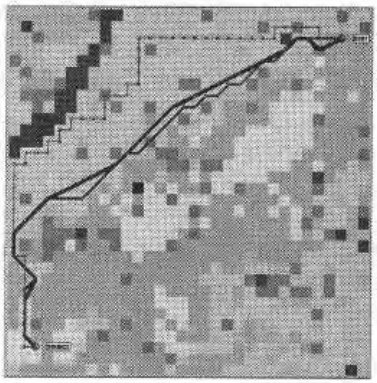


Fig. 11: 4-, 8-, and 16-connected paths.

2.3 Extended Raster Approach

The extended raster approach solves the problem of intersecting paths and possibly too expensive angles because of long edges by defining graph nodes at the sides of the raster cells instead of at the centers of the raster cells; see Fig. 12. By varying the number of graph nodes on the sides of the raster cells the number of allowed directions can be varied. Nodes in the center of raster cells (or anywhere *inside* a raster cell) are not necessary, because according to *Snell's law of refraction of light*, paths within uniform areas are always straight lines; see Subsection 3.2. Note that in the extended raster approach there are two kind of nodes: *data nodes*, which correspond to raster cells and *search nodes*, which are located on the boundaries of the raster cells.

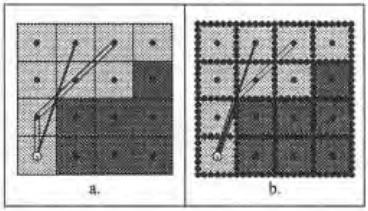


Fig. 12: Intersection and fusion.

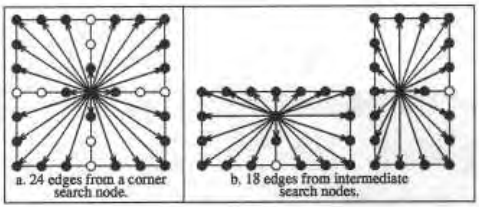


Fig. 13: 2 intermediate nodes.

The total number of search nodes n_s on a square raster of $n \times n$ raster cells (data nodes) with i intermediate search nodes on each side of a raster cell is $n_s = (2i+1)n^2+(2i+2)n+1$. In the graph, the total number of edges is $|E| = 2(3i^2+5i+2)n^2+2(i+1)n \approx 2(3i+2)(i+1)n^2$. Note that the number of directions which can be followed from each corner search node differs from the number of directions which can be followed from each intermediate search node; see Figs. 13.a and 13.b. The search graph of the extended raster approach without intermediate search nodes can be compared to the search graph of the 8-connected raster approach. The higher degree of freedom with the increasing number of intermediate search nodes i is visualized in a uniform terrain in Fig. 14, 15, and 16. However, the same improvement is also obtained in non-uniform terrains in contrast to the more-connected approach.

Another advantage of the extended raster method is that each edge is contained in exactly one raster cell. The movement cost of such an edge is calculated depending on the length and on the suitability score of only one raster cell. Horizontal or vertical moves along the raster edges use the suitability score of the cheapest cell adjacent to the edge. A drawback of the extended raster method is that the search graph gets quite dense when the number of intermediate nodes i increases. However, the search graph can be made less dense by applying Snell's law of reflection, which states that shortest paths bend towards the normal when entering a cell with higher cost; see Subsection 3.2.

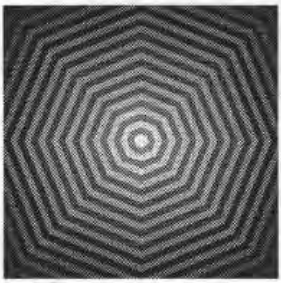


Fig. 14: $i = 0$ nodes.

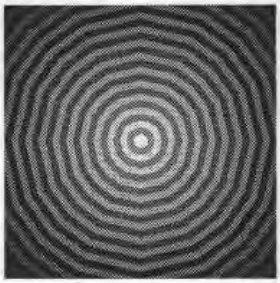


Fig. 15: $i = 1$ node.

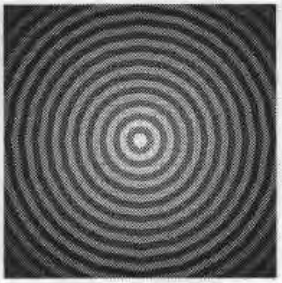


Fig. 16: $i = 2$ nodes.

2.4 Quadtree Approach

One of the drawbacks of the previously described approaches is that even in uniform areas every sample-point is stored as a separate data-object. In the *quadtree* [20] data structure, these uniform areas are grouped together. This does not only result in more efficient memory usage, but also in more efficient algorithms, because of the

smaller number of nodes and edges in the graph.

The *quadtree* is a hierarchical data structure, and is based on the successive subdivision of a raster into four equal-sized quadrants or regions. If a quadrant does not consist entirely of raster cells with the same suitability score, then it is again subdivided into quadrants, and so on. In this way regions are obtained with a uniform suitability score.

As an example of the quadtree, consider the $2^3 \times 2^3$ raster shown in Fig. 17. In Fig. 19, it is shown how this raster is subdivided into square regions. The layout of the quadtree data structure is shown in Fig. 18. All white nodes in this figure correspond with regions in the quadtree and are called *leaf* nodes. The black nodes are called *internal* nodes. The topmost node is called the *root* node. The search graph is created by connecting all adjacent (along an edge or point) cells; see Fig. 19. In order to be able to search the graph efficiently all the edges are stored explicitly.

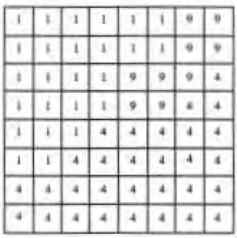


Fig. 17: Raster.

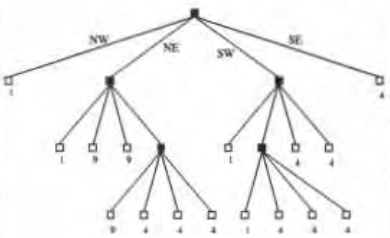


Fig. 18: Quadtree.

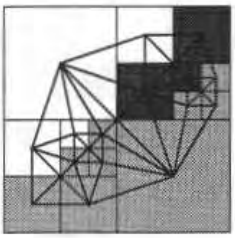


Fig. 19: Graph

Although in the worst case the number of regions in the quadtree is equal to the number of sample points (e.g. a chess board), in most cases the number of regions is much smaller. When there is not much reduction, then it is possible that the gridsize has been chosen too large in the sampling phase. One drawback of the quadtree method is that within a large quadtree node every "position" has the same distance from the source. This is visualized in an iso-time diagram as a large blob with only one color. A related problem is caused by the fact that the path has to go from center to center, which may result in non-optimal solutions in the case of large nodes; see Fig. 21. A solution for this might be a combination of the quadtree and the extended raster approach.

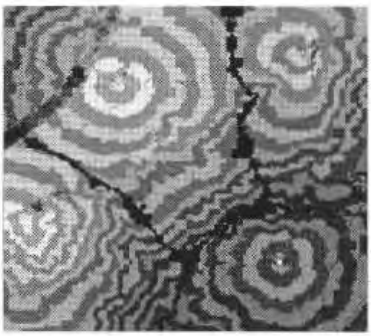


Fig. 20: Weighted Region Voronoi Diagram.

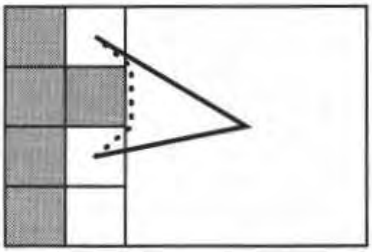


Fig. 21: Deviation error caused by large quadtree cells.

3 Vector CCM algorithm

The precision of the raster-based solutions, i.e. the quality of the returned shortest paths, depends on the fineness of the grid, and on the number of move directions permitted. The ideal solution is only found when both tend to infinity. Clearly, this is a very unpractical solution. Therefore, the vector approach, which produces an exact solution, is examined. In the vector-based movement planning problem it is now assumed that the plane is subdivided into polygonal regions. Each region (face) is associated with a positive weight $\alpha_f\,(> 0)$. However, each edge is also associated with a weight $\alpha_e\,(> 0)$ in contrast to the raster approach. These weight factors specify the cost per traversed distance in a region or along an edge. By giving the edges their own weight factor, on-road movement planning and CCM-planning can be nicely integrated. Obstacles can be seen as regions with an infinite weight. We now want to find the path with the least cost across this map given a source and a destination. The vector-based algorithm to solve the CCM planning problem (also called *weighted region problem*) is based on the article of Mitchell and Papadimitriou [15].

The first big step in the vector algorithm is to triangulate the polygonal partitioning and is described in Subsection 3.1. Some observations about shortest paths in a vector map are given in Subsection 3.2. Subsection 3.3 gives an outline of the vector CCM algorithm.

3.1 The Constrained Delaunay Triangulation

The algorithm we developed to build a *constrained Delaunay triangulation* (CDT) over a planar set of n points together with a set of non-intersecting edges has the properties that all specified points and edges can also be found in the output and it is as close as possible to the unconstrained Delaunay triangulation [18]. It runs in $O(n\log n)$ time, which is asymptotically optimal [21]. This algorithm is based on the concept of two other algorithms. The first algorithm is the unconstrained Delaunay triangulation (UDT) algorithm of Lee and Schachter [14] and the second algorithm is the CDT algorithm of Chew [2].

The input of our algorithm is a graph $G = (V, E)$ in which V is a set of n vertices and E is a set of edges, the so called *G-edges*. Two different kinds of edges appear in a CDT: G-edges, already present in the graph, and *D-edges*, created by the CDT algorithm. If the graph has no G-edges then the CDT and the UDT are the same. The graph can be thought of to be contained in an enclosing rectangle. This rectangle is subdivided into n separate vertical strips in such a way that each strip contains exactly one region (a part of the strip) which in turn contains exactly one vertex. After dividing the graph into n initial strips, adjacent strips are pasted together in pairs to form new strips. During this pasting new regions are formed of existing regions for which the combined CDTs are calculated; see Figs. 22, 23, and 24. This pasting of adjacent strips is repeated following the divide-and-conquer paradigm until eventually exactly one big strip, consisting of exactly one big region, is left for which the CDT is calculated. More details about the algorithm and its implementation can be found in [24].

3.2 Principle Properties of Shortest Paths

The following properties form the basis of the vector-based CCM algorithm and are valid for the shortest path between two points:

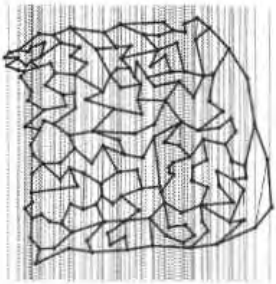


Fig. 22: Step 1, 91 strips.

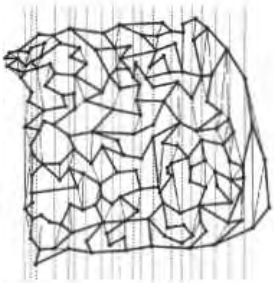


Fig. 23: Step 3, 23 strips.

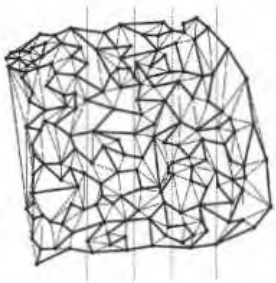


Fig. 24: Step 5, 5 strips.

1. *Within a uniform region, the shortest path does not bend.* Note that this does not mean that the shortest path between two points within the same area always is a straight line; see Fig. 25.
2. *Shortest paths do not intersect* unless they have the same distance to the source and can therefore be interchanged.
3. *On entering a new region the shortest path obeys Snell's Law of Refraction:* $\alpha_f \sin(\theta) = \alpha_{f'} \sin(\theta')$ where θ and θ' are the angles of entry and exit; see Fig. 27[3].
4. *The path entering edge e at the critical angle* $\theta_c = \arcsin(\alpha_{f'}/\alpha_f)$ *will continue along the edge* when going from a expensive to cheap region. Critical reflection occurs when the path re-enters the same face after travelling along the edge; see Fig. 25.
5. *Critical use occurs when an edge is cheaper than its surrounding faces.* It is possible that the path enters a low cost edge from face f at critical angle $\theta_c = \arcsin(\alpha_e/\alpha_f)$, then follows the edge and finally leaves the edge at critical angle $\theta'_c = \arcsin(\alpha_e/\alpha_{f'})$ through face f'; see Fig. 26.
6. *When the shortest path hits a vertex it can continue in any direction,* except that the path will not go directly back to the face where it came from.

3.3 Outline of the Algorithm

Because shortest paths so much resemble rays of light, the algorithm more or less simulates the propagation of a wavefront of light through the triangulation [15]. It is assumed that the source point is located on a vertex. If this is not the case, the triangulation can be easily modified in such a way that the source point is on a vertex. The wavefront is originated at the source point, and will initially spread in an circular way. It is propagated through the weighted triangulated map where Snell's law of refraction of light is applied each time a region boundary is crossed.

[3]To minimize the cost function $F(m_x) = \alpha_f \sqrt{(m_x^2 + s_y^2)} + \alpha_{f'} \sqrt{((t_x - m_x)^2 + t_y^2)}$ of the path from s to t trough m, the equation $F'(m_x) = 0$ has to be solved. By applying some basic goniometric calculation this can be rewritten to Snell's Law. Note that in order to find m_x the following polynom in q of degree 4 has to be solved ($q = m_x/t_x$ and $\alpha = \alpha_f/\alpha_{f'}$):

$$q^4(\alpha^2 t_x^2 - t_x^2) + q^3(2t_x^2 - 2\alpha^2 t_x^2) + q^2(\alpha^2 t_x^2 + \alpha^2 t_y^2 - s_y^2 - t_x^2) + q(2s_y^2) + (-s_y^2) = 0$$

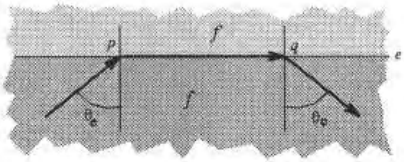


Fig. 25: Critical reflection.

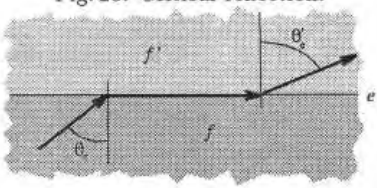


Fig. 26: Critical use.

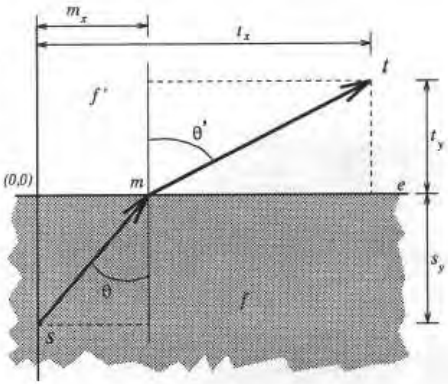


Fig. 27: Snell's Law of Refraction.

Instead of keeping track of the exact location of the wavefront, only the passing of the wavefront over certain locations is tracked. These locations are called *events*. Events are located at positions where the wavefront hits a vertex and at certain other important locations on the edges. All events which have been created, but which have not yet been overrun by the wavefront are kept in a priority queue sorted on their distance to the source. These events can be seen as the output of the algorithm because they can be used to efficiently trace back the shortest path from any point in the triangulation to the source.

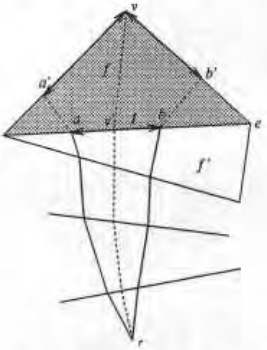


Fig. 28: Projection of interval.

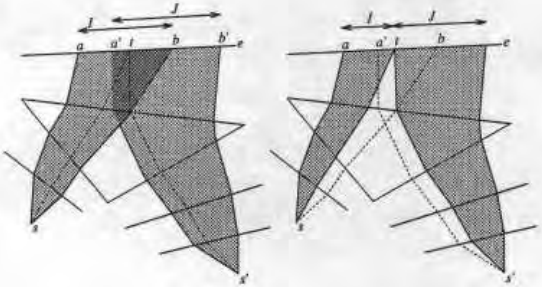


Fig. 29: Find the tie-point.

As can be seen in Fig. 28, it may be possible that a bundle of rays hits a vertex after propagation and that it need to be splitted. Another interesting situation occurs when there are two possible and efficient bundles of rays (or intervals) which reach, through a different route, a certain edge from the source, then it has to be established which point in the overlap of two intervals is equally well reached via both intervals. Between any two intervals there can only be one such point. This point, which is called the *tie-point*, can be found by performing a binary search on the overlap of the two intervals; see Fig. 29. The latter operation may be quite expensive as it involves solving a polynom of degree 4 (see Subsection 3.2) again and again for each iteration.

4 Implementation and test results

The implementation has been done in C++ with the aid of two C++ library packages: LEDA [16] for geometric data-structures and libg++ from GNU [24] for some other ADTs, e.g. the splay tree priority queue and complex numbers. The CCM-planning algorithms are connected to the Postgres/GEO++ GIS [23, 25] using a "close coupling" [7]. Full integration is planned in the near future. In Fig. 30, the large window shows the terrain and some optimal paths from three sources, the upper right window shows the iso-time diagram[4], and the lower right window shows the user interface of the analyses modules in which the parameters can be set.

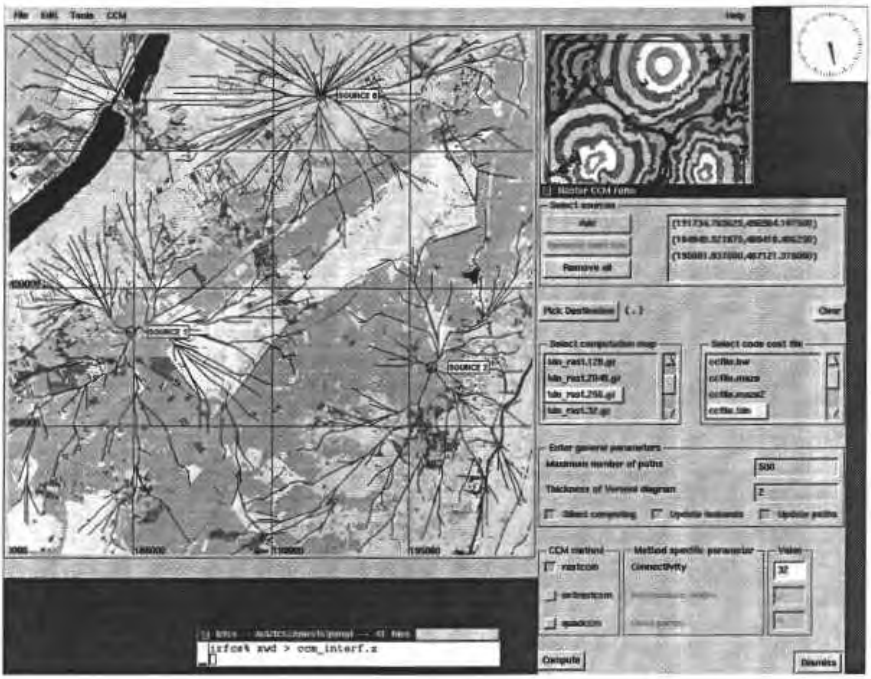


Fig. 30: User interface of CCM analysis programs.

For the tests two maps have been used; a vector data set of 181 vertices, 524 edges and 344 faces, and a raster data set of 512 × 512 raster cells representing the same area. Two sources and two destinations were chosen and after each test the average of the four source-destination paths was computed. In Figs. 31 and 32 examples are shown of paths found with a raster (32-connected) and the vector approach. The measured cost of the paths are: *more-connected raster approach:* 27.25, 22.36, 21.76, 21.58, 21.54, 21.51 (for the connectivity values 4, 8, 16, 32, 64, 128), *extended raster approach:* 22.34, 21.75, 21.62, 21.57 (for $i = 0..3$), *quadtree raster approach:* 22.47, and the *vector approach:* 21.89[5].

[4]Note that the iso-time diagram of a scene with multiple sources results in a kind of Voronoi diagram.

[5]The coordinates of the raster and vector map are slightly different. Therefore, it is dangerous to compare the length of the vector path with the raster paths.

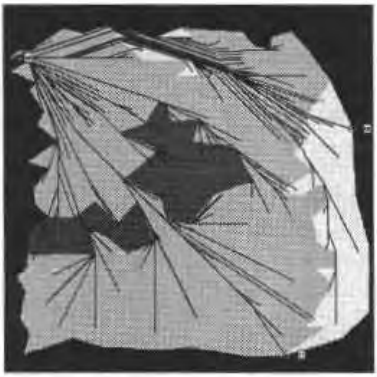


Fig. 31: Raster solution.

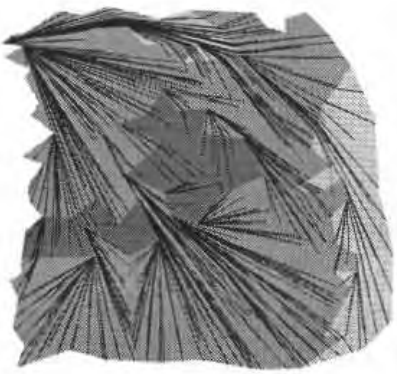


Fig. 32: Vector solution.

More test results, with different data sets and different resolutions, can be found in [24]. In these tests, CPU times, memory usage, and the quality of the paths are measured. The results clearly follow the theoretical worst case time bound of a Dijkstra search $O(e \log v)$ in a graph with v vertices and e edges. With the more-connected raster approach a connectivity rate of 16 provides the most satisfactory trade-off between the quality of paths and the CPU time. With the extended raster approach a number of intermediate search nodes of 1 provides the most satisfactory trade-off. The vector implementation turned out to be heavy resource user. However, this is not a surprise as the theoretical worst case time bound is $O(n^8)$ [15].

5 Hierarchical planning

All presented solutions require a lot of time and space to solve the problem when using realistic data sets, e.g. a 50K $\times$ 50K raster. A possible generic improvement to this is hierarchical planning: the search starts at a coarse level and is refined when required. This approach would not only reduce the required main memory, but also reduce computation time. A major drawback of most known hierarchical approaches is that they do not guarantee to find a global optimal solution. We present an algorithm that guarantees to find an optimal path. It is not an approximation or good path algorithm. A description for a raster implementation based on pyramid tree [19, 20] is given, but the same method can be applied in the vector case (based on an hierarchical constrained triangulation [26]).

The hierarchical approach is based on a hierarchical set of maps, see Fig. 33. Note that both the minimum and maximum cost values are stored in the course maps. This set starts with a coarse representation of the terrain. The next map is a more refined (detailed) version of the previous map, etc. In this way a whole list of maps exist where the last map has enough details for the application it is needed for. Assume that the shortest path from s to t has to be found.

In a hierarchical approach any cell C of a coarse version of the map will cover several different cells on a finer level. For every cell C we store the highest and the lowest resistance value of any underlying cells of the map. The following values are now calculated using a coarse version of the map:

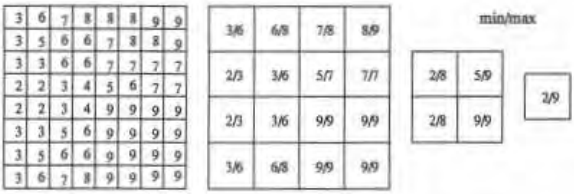


Fig. 33: Pyramid data structure with 4 levels.

1. The shortest path from s to t using the worst-case (highest) values for every part of the map. This value is an upper-bound for the real length of the shortest path from s to t.
2. For every cell C, the shortest path from s to C and the shortest path from C to t are calculated using the best-case (lowest) values. The sum of these two values is a lower-bound for the shortest path from s to t via cell C.

If the best path ever attainable via C (as described in item 2) is longer than the shortest path that was guaranteed to exist (as described in item 1), then it is sure that the shortest path will not go via cell C. In the search for the shortest path, cell C, and all its subcells, need not be considered anymore and can be discarded from memory. Next, all cells of the map which have not been discarded are replaced by a more detailed version and the whole process is repeated.

It is interesting to note that in normal cases this method discards all nodes that are outside an ellipse shaped cell around s and t. When s and t are close to each other then almost the whole map can be discarded in an early phase of the procedure, because the ellipse around these points will be small. If there is an obstacle between s and t which would force the shortest path outside this ellipse shaped cell then the cell is automatically adapted so that all possible shortest paths are included. An important issue is that only one version of the map needs to be in main memory at any time. When a part of the map is refined the refined version can be loaded from disk and replace the coarse version from memory.

Although there are similarities between quadtree approach and hierarchical approach, they are fundamentally different. The quadtree is very useful in case there are uniform parts in the terrain: its saves space and computation time. However, it can not avoid visiting a part which is unrelevant in finding the optimal path. However, the quadtree may be used to store the data pyramid efficiently.

6 Conclusion and Future work

We have implemented several raster-based and one vector-based cross country movement (CCM) planning algorithms. The raster-based algorithms have several advantages. They are relatively easy to implement and perform reasonably well. In theory however, an exact solution can only be found when both the *move-set* and the number of *search nodes* (related to the sampling resolution) in the area under consideration tend to infinity. This means that in practice only an approximation of the solution is found. The vector-based method we implemented has the advantage that it returns exact solutions. However, the performance of the vector-based method is not satisfactory.

The current raster- and vector-based approaches have major performance problems and are too slow to be used in an interactive GIS. New methods to improve the performance must be found. An extended raster approach which uses quadtree-based techniques to reduce the number of search nodes looks very promising. Further, we will implement and test the new hierarchical approach. Other types of solutions are based on using genetic algorithms [5, 13] and parallel algorithms [22].

The discrete vector approach, which avoids expensive trace-back computations to find tie-points, can be considered a combination of the extended raster approach and the vector approach. Each edge, side of a face, has a fixed number of search points which are used to create a connectivity graph. The shortest path can be found by applying the Dijkstra algorithm. This path can be enhanced locally by shifting points in such a way that on every crossing of an edge Snell's Law is obeyed [22]. Additional future work might include finding solutions for the following possible extensions to the CCM-problem:

- Find path from source to destination which passes a "variable" go-through location for generating alternative paths [3];
- Source or destination with different shapes: e.g. line, area, or multiple points;
- Path with non-zero width (e.g. a shape of 1 km width);
- Take crowdedness (multiple walkers) into account;
- Add the third dimension to the problem: the cost to travel uphill is higher than the cost to travel downhill, i.e. costs are direction sensitive (anisotropic).

References

[1] S. Baase. *Computer Algorithms Introduction to Design and Analysis*, chapter 4 and 6. Addison Wesley Publishing Company, 1988.

[2] L. Paul Chew. Constrained delaunay triangulations. *ACM*, pages 215–222, 1987.

[3] Richard L. Church, Scott R. Loban, and Kristi Lombard. An interface for exploring spatial alternatives for a corridor location problem. *Computers and Geosciences*, 18(8):1095–1105, 1992.

[4] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik 1*, 1:269–271, 1959.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, Reading, Massachusetts, 1989.

[6] M. F. Goodchild. An evaluation of lattice solutions to the problem of corridor location. *Environment and Planning A*, 9:727–738, 1977.

[7] Michael Goodchild, Robert Haining, and Stephen Wise. Integrating gis and spatial data analysis: problems and possibilities. *International Journal of Geographical Information Systems*, 6(5):407–423, 1992.

[8] P.E. Hart, N.J. Nilsson, and R. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science and Cybernetics*, SSC-4(2):100–107, 1968.

[9] B. F. Hobbs and A. H. Voelker. Analytical multiobjective decision-making techniques and power plant siting: A survey and critique. Technical Report ORNL-5288, Oak Ridge Nat. Lab., Oak Ridge Tennesee, February 1978.

[10] Dennis L. Huber and Richard L. Church. Transmission corridor location modeling. *Environmental Engineering ASCE*, 111(2):114–130, 1985.

[11] D. P. Huijsmans and A. M. Vossepoel. *Informatie in Gedigitaliseerde Beelden*, volume One: Introduction. RUL, January 1989.

[12] Scott T. Jones. Solving problems involving variable terrain. *BYTE Publications Inc*, pages 74–82, March 1980.

[13] Ir. D.W. Jonker. Genetic algorithms in planning. Technical report, FEL-TNO Divisie 2.1, June 1993.

[14] D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, 1980.

[15] J. S. B. Mitchell and C. H. Papadimitrou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *Journal of the Association for Computing Machinery*, 38(1):18–73, January 1991.

[16] S. Näher. Leda user manual version 3.0. Technical report, Max-Planck-Institut für Informatik, Im Stadtwald, D-6600 Saarbrücken, 1992.

[17] Alan V. Oppenheim, Alan S. Willsky, and Ian T. Young. *Signals and Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[18] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.

[19] Hanan Samet. *Applications of Spatial Data Structures*. Addison-Wesley, Reading, Mass., 1989.

[20] Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Mass., 1989.

[21] M. I. Shamos and D. Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on the Foundation of Computer Science*, pages 151–162, October 1975.

[22] Terence R. Smith, Cao Peng, and Pascal Gahinet. Asynchronous, iterative, and parallel procedures for solving the weighted-region least cost path problem. *Geographical Analysis*, 21(2):147–166, 1989.

[23] Michael Stonebraker, Lawrence A. Rowe, and Michael Hirohama. The implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142, March 1990.

[24] Joost van Bemmelen and Wilko Quak. Master's thesis: Cross country movement planning. Technical Report FEL-93-S205, TNO Physics and Electronics Laboratory, August 1993.

[25] Tom Vijlbrief and Peter van Oosterom. The GEO system: An extensible GIS. In *Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, South Carolina*, pages 40–50, Columbus, OH, August 1992. International Geographical Union IGU.

[26] J. M. Ware and C. B. Jones. A multiresolution topographic surface database. *International Journal of Geographical Information Systems*, 6(6):479–496, 1992.

SPATIAL AND TEMPORAL VISUALISATION OF THREE-DIMENSIONAL SURFACES FOR ENVIRONMENTAL MANAGEMENT

Malcolm J. Herbert and David B. Kidner

Department of Computer Studies
University of Glamorgan
Pontypridd, Mid Glamorgan
United Kingdom CF37 1DL

ABSTRACT

In geographical modelling, various triangulation techniques are used to model three-dimensional data, to provide computer-based visualisation. This now forms an important component of many geographical information and three-dimensional modelling systems.

There are an increasing number of data sets that now provide information about a particular model with a temporal element, with a repeated number of measurements made at a particular site. The resultant images make up a series of snapshots of the model, but may not represent the significant points in the life of the model. We propose two methods that use serial section reconstruction techniques, which can be used to construct three-dimensional models and to interpolate new models at other moments in time.

A prototype system has been developed and is illustrated with data provided by the Countryside Council for Wales of a sand-shingle spit. This has been surveyed at regular intervals since 1981, to examine the effects of coastal management and climatic conditions on the spit.

INTRODUCTION

A three-dimensional object can be commonly represented by a set of serial sections, where contours define the intersection with a plane (or section), and the surface of the object. On a conventional map, contours of varying heights (or depths) appear together on a two-dimensional drawing to give an indication of the shape and form of the surface. Other disciplines, such as medical imaging (Rhodes 1990), geology, and palaeontology (Ager 1956), use serial sectioning to produce a set of two-dimensional images of an object. This allows the observer to view otherwise hidden features.

Various techniques have been developed to create computer-based three-dimensional models from these sections (Ekoule, Peyrin and Odet 1991, Meyers, Skinner and Sloan 1992).

This paper illustrates how two surface reconstruction techniques can be used to display three-dimensional geographical data. It also discusses how these methods can be adapted to carry out temporal interpolation between existing three-dimensional models.

SERIAL SECTION RECONSTRUCTION.

Introduction.

The problem of reconstructing three-dimensional surfaces from two-dimensional sections can be broken down into a number of sub-problems (Meyers, Skinner and Sloan 1992), (see Figure 1). The *correspondence problem* is concerned with specifying the underlying topology of the object and the correct connectivity between contours in adjacent sections. The *tiling problem* is that of finding the best set of triangulated facets to define the surface between a pair of connected contours. The *branching problem* occurs when the connectivity between contours is not a simple one-to-one scenario and the *surface fitting problem* fits a smooth surface to the resultant triangulation, to facilitate visual inspection of the reconstructed object.

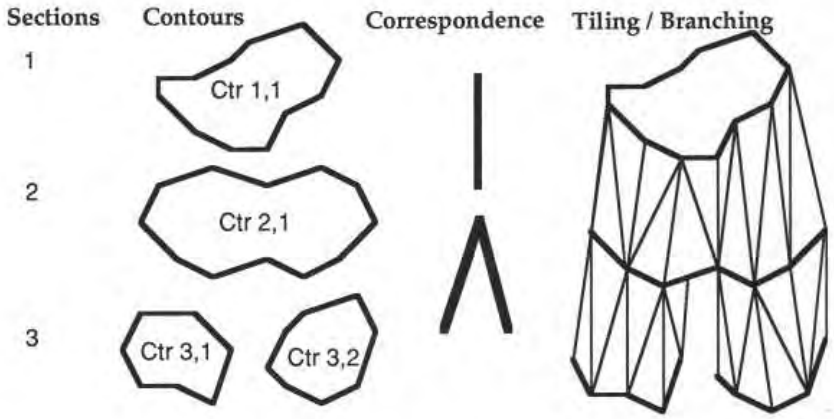


Figure 1. Serial Section Reconstruction Sub-Problems.

Tiling and Branching Problems.

The tiling problem was the first sub-problem to be addressed in detail by researchers. There are a number of solutions, which decompose the problem from the section level to the contour level and then 'stitch' together a triangulated surface between a pair of contours (Keppel 1975, Fuchs, Kedem and Uselton 1977, Christiansen and Sederberg 1978). Refinements have been made to these original methods to enable the handling of more complex scenarios (Sloan and Painter 1988, Ekoule, Peyrin and Odet 1991), and to resolve the branching problem (Meyers, Skinner and Sloan 1992). A solution based on three-dimensional Delaunay triangulation (Boissonnat 1988), tackles the problem at the higher, section level and therefore has the advantage of not requiring the prior resolution of the correspondence problem or the need to treat branching as a special case. This method has been used successfully for complex objects in medical imaging (Lozanoff and Deptuch 1991), but it has not received the same attention as the contour-level methods.

Correspondence Problem.

The correspondence problem has received less attention from researchers than the tiling problem. This is because it's importance is not as great when dealing with well sampled data sets, as are encountered in medical imaging. Reconstruction algorithms usually rely on user input (Giertsen, Halvorsen and Flood 1990), or simple metrics (Ekoule, Peyrin and Odet 1991) to resolve the problem. More complex data sets have been solved automatically (Shinagawa and Kunii 1991, Meyers, Skinner and Sloan 1992) using graph based approaches, whilst information about the relationships between contours in the same section have been used to help deduce a solution (Haig, Attikiouzel and Alder 1991).

Surface Fitting Problem.

Surface fitting is used to improve the quality of the final three-dimensional image. Usually the triangulated facets will not provide an adequate result, as the contours do not necessarily sample the original object too well. A common method is to fit parametric surface patches based on the vertices of triangulation (Farin 1990).

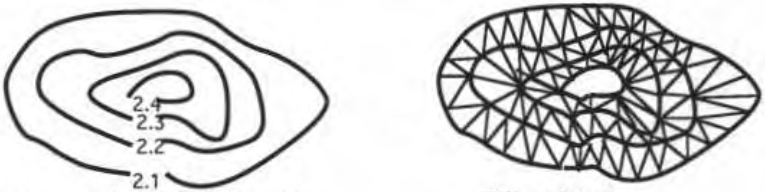
THREE-DIMENSIONAL RECONSTRUCTION.

A set of programs has been developed, primarily to reconstruct palaeontological specimens from serial sections (Herbert 1993). These have also been applied to a number of other geoscientific areas. These include contour data from land surveys as well as geological data. The paper concentrates on the tiling and branching algorithms and their potential for temporal interpolation, but notes that the successful resolution of the correspondence problem plays an important part in producing complex

images. This can be attributed to the nature of the data and lack of sampling in many geoscientific data sets. Two different approaches to the tiling and branching modules have been implemented and are described below.

Contour-Level Method.

The first is a contour-level algorithm based on the heuristic proposed by Christiansen and Sederberg (1978). It produces a series of triangular facets between adjacent contours, which have been matched after the resolution of the correspondence problem (see Figure 2) . Changes to the original algorithm allow the tiling of extremely concave contours and the detection and resolution of complex branching scenarios (Herbert 1993).



Contours - with Heigth(Depth). Tiling Solution.

Figure 2. Sample Contours, with a tiling solution.

Section-Level Method.

The second method, a section-level routine based on that of Boissonnat (1988), has been developed, both to compare it with the contour-level routine and also to evaluate it's potential with complex branching scenarios. Here surfaces are constructed between all the contours that appear in an adjacent section pair simultaneously. The method consists of two stages. The first stage constructs a two-dimensional constrained Delaunay triangulation (CDT), based on the contour vertices and edges, in both of the sections. The second stage uses these triangulations and their Voronoi duals to construct a set of tetrahedra that lie between the two sections. The final surfaces are obtained by stripping away unwanted tetrahedra from the three-dimensional object, and extracting the external triangular facets. A number of improvements to the Boissonnat method have been made, including using an iterative method to calculate the two-dimensional CDT (De Floriani and Puppo 1988) . Other enhancements allow the reconstruction of objects with holes and the ability to restrict certain surfaces between contours, allowing an object to be made up of separate components.

By adapting the routines for three-dimensional reconstruction, it is possible to build another model at any time throughout the history of measurement of the site. The methods can be used to interpolate the framework of a new model, by deducing new contour sets, based on two existing models. For a pair of three-dimensional models, one at time t1 and the other at time t2, a number of new contour sets can be interpolated, whose times lie between those of t1 and t2 (see Figure 3).

Figure 3. Changes in Contours through time.

Certain features must be present in a set of data, to make temporal interpolation possible, using the methods described below.

1. The contours are recorded at the same heights and intervals throughout the sequence of surveying.
2. The surveys are recorded in some form of standard coordinates and contain reference points that appear in all of them. This enables the models to be aligned.

This situation which allows temporal interpolations to be made can be defined as follows. There exists a number of surveys or measurements, taken at time $t_1 \ldots t_n$. For each survey, there are a set of sections $s_1 \ldots s_n$, which contain a set of contours $c_1 \ldots c_n$. Only one section will exist at a certain height (depth) and there will be one or more contours in each of the sections. A contour vertex is defined as a four-dimensional coordinate (x,y,z,t), where x and y are longitude and latitude, z is the height (depth) value of the section and t is the time.

Contour Level Method.

This method requires the correspondence between pairs of contours in the adjacent models, which have the same height (ie in the same section), to be deduced first. The tiling and branching algorithm can then follow this to produce a triangulation between the contour pair. The height values for the contour are ignored during this process and this is replaced with the time value. From this triangulation, a new contour, with a time value lying between the times of the two original contours can be extracted (see Figure 4). This contour will be at the same height (depth) value as the original contours.

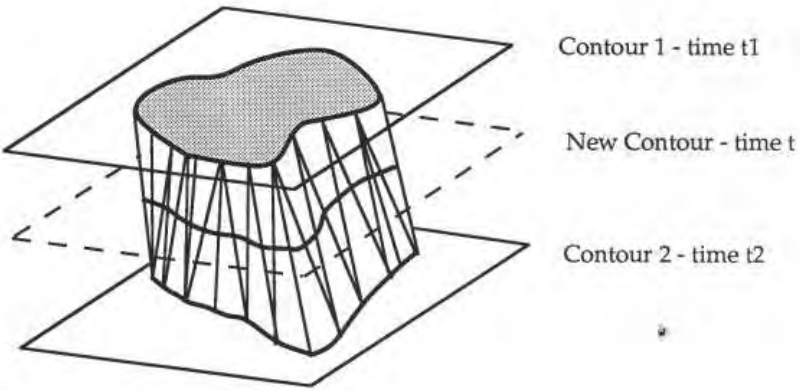


Figure 4. New Interpolated Contour.

Hence, a set of contours at the new time, can be extracted from sets of contour pairs, which are at various height levels in the original models. A three-dimensional model can then be built from these contours at this new time. Whilst this process is relatively simple for straightforward sets of data, as with the three-dimensional process, there is a need for an automatic correspondence solution when reconstructing more complex data sets.

Section-Level Method.

This method also requires the matching of sections of the same height value in a time adjacent pair of three-dimensional models, but does not require the correspondence to be resolved. As with the three-dimensional reconstruction section-level method, branching situations (see Figure 5) are handled by the routine automatically. Again the height (depth) level is replaced with the two time values, and a new cross section is calculated at the new interpolation time.

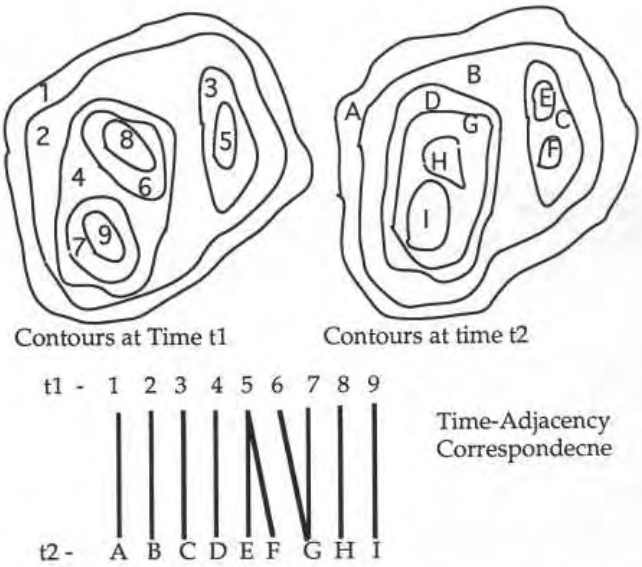


Figure 5. Time Adjacency Correspondence, with Branching.

This method may be more suitable for modelling complex situations, as it does not require the correspondence to be resolved beforehand and since it is polyhedra-based it can handle objects with more complex topology (Lozanoff and Deptuch 1991).

MODELLING FOR ENVIRONMENTAL MANAGEMENT

An area of the Gronant Dunes, near Prestatyn, North Wales, United Kingdom, a designated Site of Special Scientific Interest (SSSI), has been surveyed regularly by the Countryside Council for Wales (CCW) since 1981. This is in order to monitor topographic changes to the site and to assess how the nesting birds are being affected, especially since groynes have been installed further along the coast. The prime motivation for the computer-based modelling was visualisation and the opportunity to experiment with the extension to serial section reconstruction to enable some form of temporal interpolation. The contours from the maps produced each year were digitised and three-dimensional surfaces reconstructed using both of the contour and section-level models. The three-dimensional images benefitted from the use of false height values for the contours to help

visualisation, as well as the use of surface fitting in the form of Gouraud shading.

The interpolation and creation of new models at intermediate points in time can show the rate of change to the site and also the way in which it changes. This data when used in conjunction with other information, such as climatic conditions, can be used to predict effects on the site and plan for future management. It provides additional ways in which the data can be analysed and presented.

CONCLUSION.

With both the three-dimensional reconstruction and the temporal interpolation, the section-level or Boissonnat method is capable of handling more varied data sets, without the need to solve complex correspondence situations. However, the contour-level method is more flexible, in that by defining the correspondence solution prior to tiling , the underlying topology of the final three-dimensional object can be determined.

The data set that has been presented is fairly straightforward and there is no doubt further work is required to model more complex geoscientfic data sets. Firstly there is potential for the section-level or Boissonnat method to interpolate between two complete three-dimensional model pairs without decomposition to the section level. The surface facets could be used to build a set tetrahedra directly, with new intermediate surfaces being extracted from them. Another area involves the development of more accurate solutions to the correspondence problem, so enabling the contour-level method to be more effective with highly complex data sets, both in three-dimensional reconstruction and temporal interpolation.

ACKNOWLEDGEMENTS

The authors wish to thank Mike Hesketh of the Countryside Council of Wales for allowing the use of the Gronant Dunes data set. Malcolm J. Herbert is supported by a Studentship from the Scientific and Engineering Research Council.

REFERENCES

Ager D.V. 1965, Serial Grinding Techniques, in Handbook of Palaeontological Techniques ed. Kummel B. and Raup D. pp 212-224.

Boissonnat J-D. 1988, Shape Reconstruction from Planar Cross Sections: Computer Vision, Graphics and Image Processing, Vol. 44 pp1-29.

Chrsitiansen H.N. and Sederberg T.W. 1978, Conversion of Complex Contour Line Definitions into Polygonal Element Mosiacs: A C M Computer Graphics, Vol. 12 (3) pp187-192.

De Floriani L. and Puppo E. 1988, Constrained Delaunay triangulation for multiresolution surface description: IEEE Computer Society Reprint (Washington DC:Computer Society Press). (Reprinted from Proceedings Ninth IEEE Conference on Pattern Recognition, Rome, November 1988).

Ekoule A.B., Peyrin F.C. and Odet L. 1991, A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours: ACM Transactions on Graphics, Vol 10 (2) pp182-199.

Farin G. 1990, Curves and Surfaces for Computer Aided Geometric Design - a practical guide, Academic Press.

Fuch H., Kedem Z.M. and Uselton S.P. 1977, Optimal Surface Reconstruction from Planar Contours: Communications of the ACM, Vol. 20 (10) pp693-702.

Ganapathy S. and Dennehy T.G. 1982, A New General Triangulation Method for Planar Contours: ACM Computer Graphics , Vol. 16 (3) pp69-75.

Giertsen C., Halvorsen A. and Flood P.R. 1990, Graph-directed Modelling from Serial Sections: The Visual Computer, Vol. 6 pp284-290.

Haig T.D.,Attikiouzel Y. and Alder M. 1991, Border Marriage : matching of contours of serial sections: IEE Proceedings-I , Vol. 138(5) pp371-376.

Herbert M.J. 1993, Three-dimensional Reconstruction and Analysis of Natural Scientific Data: Transfer Report - Department of Computer Studies, University of Glamorgan p38.

Lozanoff S. and Deptuch J.J. 1991, Implementing Boissonnat's Method for Generating Surface Models of Craniofacial Cartilages: The Anatomical Record, Vol. 229 pp556-564.

Meyers D., Skinner S, and Sloan K.R. 1992, Surfaces from Contours: ACM Transactions on Graphics, Vol. 11 (3) pp228-258.

Rhodes M. 1990, Computer Graphics in Medicine: IEEE Computer Graphics and Applications, March pp20-23.

Shinagawa Y. and Kunii T.L. 1991, Constructing a Reeb Graph Automatically from Cross Sections: IEEE Computer Graphics and Applications, November pp44-51.

Sloan K.R. and Painter J.,1988, Pessimal Guesses May be Optimal : A Counter Intuitive Search Result: IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 10 (6) pp949-955.

COLOR REPRESENTATION OF ASPECT *AND* SLOPE SIMULTANEOUSLY

Cynthia A. Brewer and Ken A. Marlow
Department of Geography
San Diego State University
San Diego, CA 92182
cbrewer @ sciences.sdsu.edu

ABSTRACT

Systematic variations in all of the perceptual dimensions of color were used for the aspect/slope color scheme discussed in this paper. Eight aspect categories were mapped with an orderly progression of colors around the hue circle. Three slope categories were mapped with differences in saturation, and near-flat slopes were mapped with gray for all aspects. Lightness sequences were built into both the aspect and slope progressions to produce systematic perceptual progressions of color throughout the scheme (among all 25 map categories). These lightness sequences approximated relief shading and were crucial for visualization of the shape of a continuous terrain surface. Use of the HVC color system allowed easy specification of 25 colors that accurately met these interlinked color-appearance criteria. Our aspect/slope color scheme allows accurate terrain visualization while simultaneously categorizing the surface into explicit aspect and slope classes represented within a single, two-dimensional, planimetric map.

INTRODUCTION

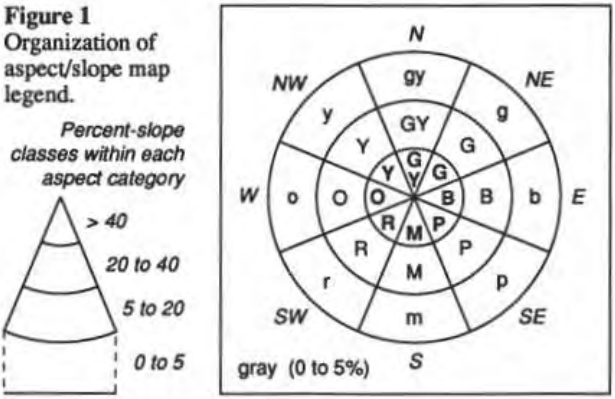
In 1990, Moellering and Kimerling presented a method of coloring aspect categories for the visual presentation of terrain or other statistical surfaces. They named this process MKS-ASPECT (GIS World 1991). Terrain data were classed into aspect categories that were appropriately symbolized with different saturated hues. In addition to using hue to symbolize qualitative aspect categories, inherent differences in the lightness of these saturated hues were ordered to produce lightness gradients that approximated relief shading. Thus, yellow, the lightest hue, was used for northwest-facing slopes and dark blue was used for southeast-facing slopes. The remaining categories were colored with saturated hues from an ordered progression around the hue circle (yellow, green, cyan, blue, magenta, red, orange).

In this paper, we describe an improvement on the MKS-ASPECT color scheme that we developed with the aid of the Tektronix HVC color system (Taylor and others 1991). In addition to aspect categories, we classed our terrain data into slope categories and represented slope with differences in the saturation of each aspect hue (Figure 1). We present the development of our aspect/slope color scheme in the first section of this paper. In the second section, we provide an outline of the ways we processed terrain data with ARC/INFO GIS software. The importance of color-system properties to development of this color scheme will be described more fully at the Auto Carto 11 conference where color examples can be presented. The full-color aspect/slope map will also be presented at the conference.

COLOR SCHEME DEVELOPMENT

The maximum saturations of hues that can be produced on a CRT vary widely in both their lightness and saturation. The wide, screened line plotted on the Hue/Chroma graph in Figure 2 shows maximum saturations varying from 93 Chroma for red to 41 Chroma

Figure 1
Organization of aspect/slope map legend.



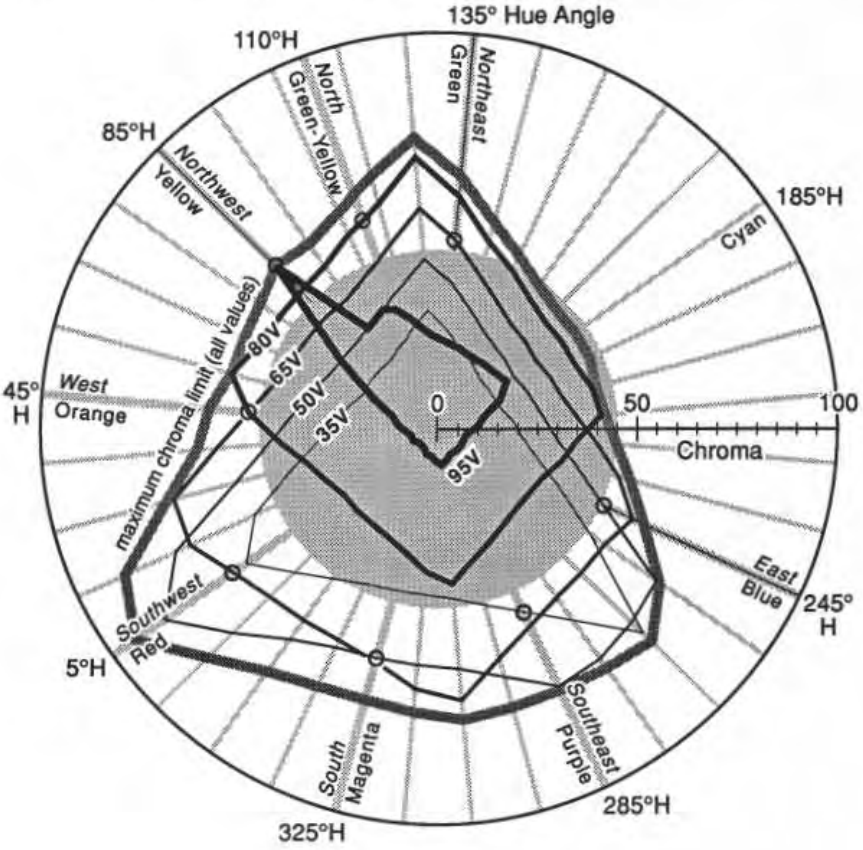
Hue initials label aspect/slope categories (see Appendix):

Lower-case letters (b, for example) show positions of desaturated colors of each hue.

Plain upper-case letters (B) are medium-saturation colors.

Bold upper-case letters (**B**) are maximum-saturation colors at the center of the legend.

Figure 2
Hue and Chroma dimensions of HVC arranged as a polar-coordinate system. Maximum Chromas at selected Value levels are plotted, and point symbols mark positions of colors selected for maximum-slope categories of eight aspects.



for cyan. The lightness of colors of maximum saturation vary from 95 Value for yellow (at 85° Hue) to 38 Value for purple-blue (at 265° Hue), which is 180° around the Hue circle from yellow.

A focus on the characteristics of colors of maximum saturation, however, omits a much greater variety of choices that are lighter, darker, and less saturated than each saturated color and that fill out the whole of the three-dimensional HVC color space. We used a sample of these colors to map aspect and slope simultaneously. Maximum Chromas of hues at specific lightness levels (35, 50, 65, 80, and 95 Value) are plotted in Figure 2 with progressively thicker lines. Figure 2 provides a schematic representation of HVC color space for the black-and-white medium of this proceedings.

Anchor Hues for Aspect Colors

To symbolize aspect and allow relief visualization, yellow was used for northwest aspects, as with the MKS-ASPECT scheme (the map had a north-up orientation). The yellow at 85° Hue has the highest Value (95) of all hues at maximum Chroma. Thus, 85° Hue was the initial anchor on the hue circle for this design problem. With this anchor, the easiest solution would have been to progress in 45° increments around the hue circle for maximum contrast between eight aspect categories. Unfortunately, two other restrictions on the solution interfered with this simple strategy.

First, all colors used to represent maximum-slope categories needed sufficient saturation that differentiable medium- and low-Chroma colors were available within each hue. We set the usable minimum at 45 Chroma, which provides increments of at least 15 Chroma units between slope classes: 0 to 5-percent slope (near flat with no hue), 5 to 20, 20 to 40, and greater than 40-percent slope (Figure 1). The gray circle in the center of Figure 2 tints all color positions below 45 Chroma (choice of this limit was a subjective decision based on iterative alterations of the color scheme). Figure 2 shows that the yellow at 85° is the only hue that attains a Chroma over 45 at the Value of 95. In addition, no colors between 155° and 225° Hue have Chromas greater than 45. Cyans are in this range. Moellering and Kimerling (1990) also had difficulty with maximum-saturation cyans because they were too light. The more fundamental problem with these hues is that they have limited saturation at all lightness levels.

The second restriction on color selection was that colors must progress from light to dark in both directions around the hue circle as aspect categories progress from northwest to southeast (NW-N-NE-E-SE and NW-W-SW-S-SE). We set the Value of the darkest color, for southeast aspects, at 35 because that provided four increments of 15 Value units between the darkest and lightest aspect colors (35, 50, 65, 80, 95). A Value of 35 also provided a reasonable range of hue choices with adequate Chroma for the darkest southeast category (Hues 255° to 295° have Chromas greater than 45 at 35 Value; Figure 2). Moellering and Kimerling (1990) attempted to match lightness measures to a cosine function (a common basis for automated relief shading), but we felt it was more important to produce equal and generous differences in lightness between all five categories within the two lightness progressions from northwest around to southeast. This strategy aided discrimination of aspect classes at all slope levels.

To be consistent with the chosen Value progression, east aspects were represented with colors one step lighter than the darkest southeast class at 35 Value. Moving toward cyan from the purple-blue side of the hue circle, 245° is the last hue with a Chroma greater than 45 at the next lightness step of 50 Value. Continuing around the hue circle, 135° is the first Hue angle at which the next lightness step (65 Value) is available at adequate saturation for the northeast maximum-slope category. A gap of 110° across the cyans exists, within which there are no useful colors available as saturated hues (use of these

cyans at the desired lightness levels produced alarmingly grayish colors in the midst of a group of vibrant saturated hues for other aspects). Thus, the green at 135° and the cyan-blue at 245° provided two more anchors for aspect-hue selection in addition to yellow at 85°. The limited usefulness of the cyans is a restriction dictated by the limits of color CRT technology rather than a flaw in the HVC color specification system.

Selection of Remaining Aspect Hues

Use of 265° Hue for the darkest southeast aspects would have been a logical choice because it is the hue that has maximum saturation at low lightness and it is complementary to, or opposite, yellow on the HVC hue circle. However, it is only 20° from the anchor cyan-blue at 245° (discussed above). The higher saturations available in the purple-magenta-red-orange range of the hue circle provided good flexibility for color selection. Spanning the range from yellow at 85° to cyan-blue at 245° with equal Hue-angle increments produced color differences of 40° that were approximately equal in appearance (because HVC approximates a perceptually scaled color system). We felt that using large hue contrasts when possible was more important than maintaining complementarity.

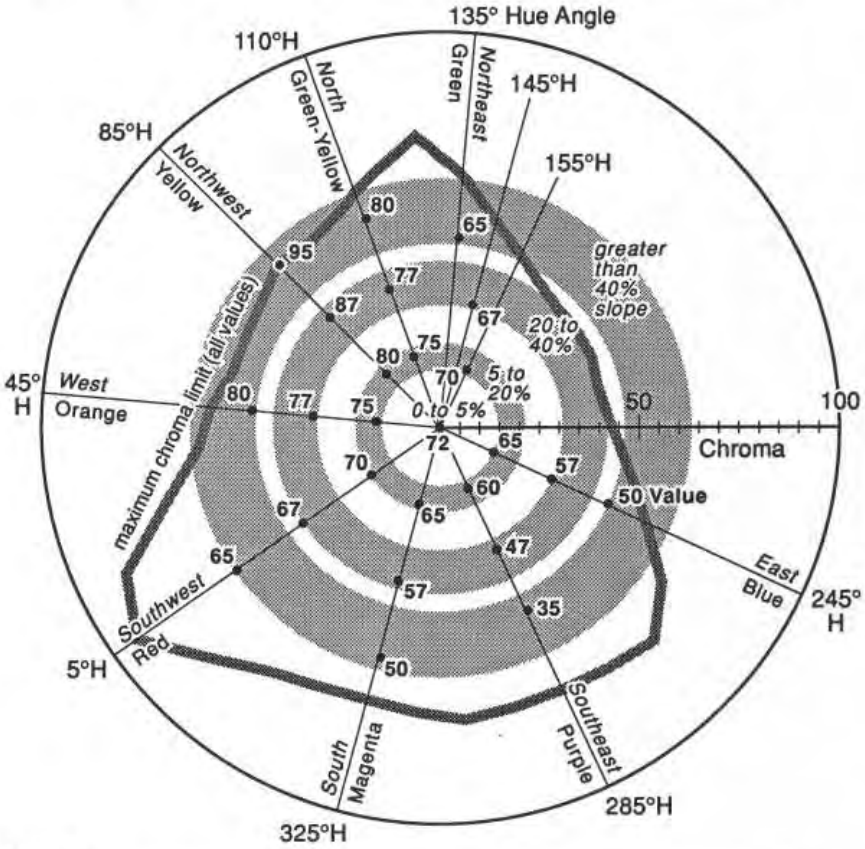


Figure 3
Hue, Value, and Chroma of all aspect/slope map categories. Bold numbers within the Figure list Values of map colors. Ranges of Chroma for each slope class are represented by tinted rings.

Selection of hues at 40° increments also provided a good approximation of unique red at 5° and the other hues were reasonable representatives of hue names (orange at 45°, magenta at 325°, and purple at 285°). Note that Chroma was maintained above 45 with a Value progression from 35 to 95 for the maximum-slope categories (Figures 2 and 3). Moellering and Kimerling (1990) deleted a magenta range of hues as they worked with HLS because saturated magentas were too light for their south-aspect category. Working with a more sophisticated color system, it is apparent that magentas of the desired darkness have ample saturation for coloring steep slopes (unlike dark cyans).

The final hue-selection problem was choice of a green-yellow of 65 Value for north aspects. This color was squeezed between the yellow anchor at 85° Hue and the green anchor at 135° (both discussed above). An equal Hue increment of 25° assigned 110° Hue to north aspects. Although this hue increment produced less visual contrast between the three adjacent aspect categories (NW-N-NE), the reasons for maintaining 85° and 135° Hue categories outweighed this deficiency. To assist discrimination of the hues at lower saturations, we used a small Hue shift from 135° to 145° and 155° for lower slope categories (Figure 3). Discrimination of these aspect categories was also aided by lightness contrasts (note the bold Value numbers within Figure 3).

Slope Category Coloring

In addition to adding slope categories to the aspect color scheme, we also added a gray category of near-flat slopes (0 to 5 percent) that was represented without hue to encourage the interpretation that these areas had no aspect. Chang and Tsai (1991) have recommended that "flat" areas should be included as an aspect class on terrain maps because they found that errors in automated aspect calculations are concentrated in areas that are generally flat with minor landforms. As slopes became flatter within each aspect category, we reduced Chroma and set Value levels closer to 72, the flat-area Value. These dual progressions in both Value and Chroma with slope are graphed in Figure 3.

Initially, flat areas were represented with a 65-Value gray, the same middle Value as southwest and northeast aspects. This choice eliminated value contrasts between the slope categories for these directions, which caused slope colors to be difficult to distinguish. In the final scheme, the near-flat areas were represented with Value 72, which was midway between Values 65 and 80 that were used for maximum-slope categories. This decision produced lightness contrasts between all adjacent map classes (Figure 3).

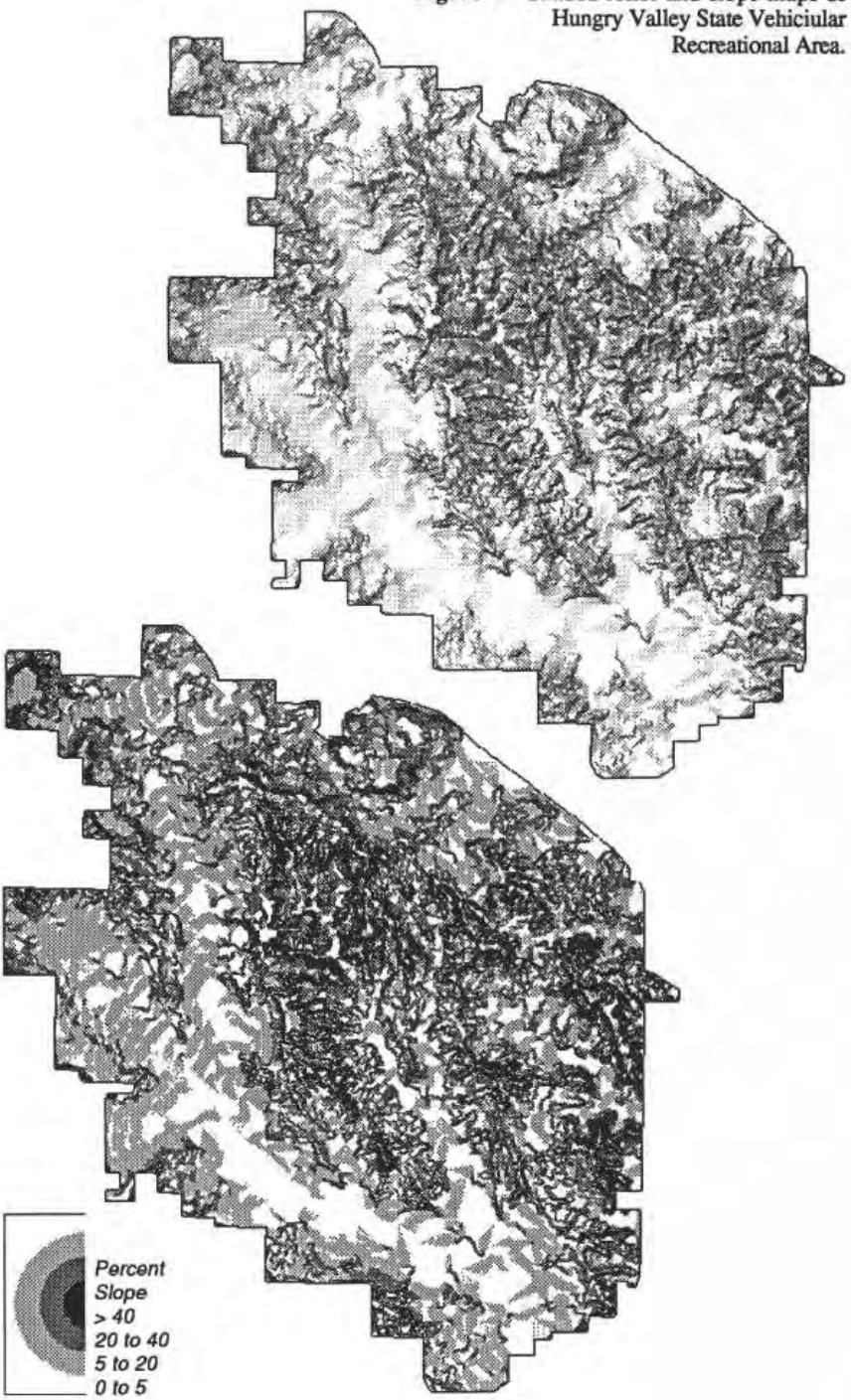
ARC/INFO TERRAIN PROCESSING PROCEDURES

The aspect/slope map used to develope the color scheme was generated with a Sun/Sparc2 workstation using the TIN, GRID, and ARCPLOT modules of ARC/INFO (version 6.1). The map was developed from a database of digitized topography of Hungry Valley State Vehicular Recreation Area, a park of 19,000 acres that is located approximately 60 miles north of Los Angeles, California. Park elevations range from 3000 to 6000 feet. Figure 4 shows shaded relief and slope maps of the park at reduced size to illustrate the terrain characteristics. The methods used to classify the terrain data for aspect/slope mapping are outlined in the sub-sections that follow.

TIN Creation

Initial coverages, or data layers, required for the project were a line coverage of digitized topography that extended beyond the park boundaries and a polygon coverage of the park boundary (HVTOPO and HVBOUND are corresponding file names in the command lists below). Development of the map required creation of a triangulated irregular network, or

Figure 4 Shaded-relief and slope maps of Hungry Valley State Vehiciular Recreational Area.



TIN, from the digitized topography. Creation of the TIN (HVTIN) was an intermediate step in the conversion of linear (contour) and point (spot) elevation values to a regular grid or lattice of elevations. Subsequent calculations of slope and aspect were made using the regular grid of elevations. Commands used to create the TIN follow:

```
ARC: DISPLAY 9999 3
ARC: CREATETIN HVTIN
CREATETIN: COVER HVTOPO LINE SPOT
```

To verify the accuracy of the TIN coverage as a surface model, a line coverage (HVARC) was generated with the TIN module and it was clipped using the park boundary. The line coverage aided detection of errors by allowing observation of inappropriate triangles:

```
ARC: TINARC HVTIN HVARC LINE
ARC: CLIP HVARC HVBOUND HVARCCLP LINE
ARC: ARCPLOT
ARCPLOT: MAPEXTENT HVTOPO
ARCPLOT: ARCS HVARCCLP
```

Similarly, assessment of the TIN accuracy was made by generating a contour map from the TIN and comparing it to the original topography coverage. Problems in the surface representation may require regeneration of the TIN after point elevations, hard breaklines, and soft breaklines are added to the original topography coverage to improve representation of surface features. The verification contour map (HVTINCON) was generated with the following command:

```
ARC: TINCONTOUR HVTIN HVTINCON 40 440 SPOT # 1
```

ARCPLOT was then used to display the resulting contour map:

```
ARCPLOT: MAPEXTENT HVTOPO
ARCPLOT: ARCS HVTINCON
```

Elevation Grid Creation

A lattice model (HVLATTICE) of the TIN (HVTIN) was generated using linear interpolation. Generation of the lattice produced a regular grid of elevation values that allowed representation and manipulation in ARC/INFO's GRID module. Default settings were accepted for the interpolation, and lattice sampling was set to 10 coverage units (meters).

```
ARC: TINLATTICE HVTIN HVLATTICE LINEAR
        enter (accepting defaults)
        enter
        enter
DISTANCE BETWEEN MESH POINTS: 10
```

A grid coverage with a value-attribute table of integer values was developed from the lattice:

```
ARC: GRID
GRID: HVGRID1 = INT(HVLATTICE)
```

Grid generation was verified with the following two commands:

```
GRID: LIST HVGRID1.VAT
GRID: DESCRIBE HVGRID1
```

A visual check of the appropriateness of the distance between sample elevation points and of the linear interpolation method was made by generating and displaying an analytical hillshade model of the grid (the shaded relief map is shown at reduced size in Figure 4):

```
GRID: HILLSHADEFULL = HILLSHADE(HVGRID1, 315, 70, ALL)
GRID: AP MAPEXTENT HVTOPO
GRID: GRIDPAINT HILLSHADEFULL VALUE IDENTITY WRAP GRAY
```

The resulting display of the grid indicated a reasonable resolution, and the grid was clipped to eliminate values outside the study area:

```
ARC: LATTICECLIP HVLATTICE HVLATTICECLP
```

GRID was again enabled to convert the floating-point grid (HVLATTICECLP) to an integer grid:

```
GRID: HVGRID2 = INT(HVLATTICECLP)
```

Aspect and Slope Map Creation

Aspect and slope grids were generated. For the slope calculations, coverage z units (vertical units) were converted from feet to meters to use the same units as horizontal measurements.

```
GRID: HVASPECT1 = ASPECT(HVGRID2)
GRID: HVSLOPE1 = SLOPE(HVGRID2, .3048, PERCENTRISE)
```

Values in both grids were converted to integers:

```
GRID: HVASPECT2 = INT(HVASPECT1)
GRID: HVSLOPE2 = INT(HVSLOPE1)
```

Generation of the aspect grid produced values ranging from –1 to 359 with 0 representing north. Cells that had a value of –1 represented flat areas with no aspect. These flat areas, which also had no slope, were represented by 0 in the slope grid.

A reclassed aspect grid (HVASPECT3) was created that was limited to eight classes corresponding to eight compass directions (N, NE, E, SE, S, SW, W, NW). The grid was reclassified by accessing an ASCII remap table (ASPECTREMAP, Table 1) that was previously created using the Sun system's text editor.

```
GRID: HVASPECT3 = RECLASS(HVASPECT2, ASPECTREMAP, DATA)
```

The slope grid was also reclassed using an ASCII remap table (SLOPEREMAP, Table 1). The remap table produced four slope categories with assigned values of 10, 20, 30 and 40.

```
GRID: HVSLOPE3 = RECLASS(HVSLOPE2, SLOPEREMAP, DATA)
```

Aspect Map Display

A preliminary set of colors, or SHADESET, was developed using the HLS color system for display of the eight-direction reclassification (HVC was not available within ARC/INFO). The SHADESET was produced using the custom menu in SHADEEDIT, ARCPLOT's symbol editor. Before accessing SHADEEDIT, reduction of screen-color flashing during simultaneous display of dynamic and static colors was accomplished by specifying the allowable numbers of dynamic and static colors as follows:

```
ARC: DISPLAY COLORMAP 215 60
ARC: ARCPLOT
ARCPLOT: SHADEEDIT
```

Once the preliminary SHADESET was developed (HLS.SHD), the aspect map was displayed:

```
ARCPLOT: SHADESET HLS.SHD
ARCPLOT: GRIDSHADES HVASPECT3 VALUE NOWRAP
```

Combination of Aspect and Slope Maps

Values of the slope and aspect grids were added to generate a new grid with cell values that contained their summation:

```
GRID: HVPLUS1 = HVSLOPE3 + HVASPECT3
```

The addition produced unique cell values for all pairings of the four slope categories with the eight aspect categories. An additional reclassification set all aspects in the lowest slope category to the same value (19) as the flat areas (PLUSREMAP, Table 1).

```
GRID: HVPLUS2 = RECLASS(HVPLUS1, PLUSREMAP, DATA)
```

An additional ASCII remap table (25COLORREMAP, Table 1) was used to convert these new values to numbers equaling the SHADESET symbol designations (1 to 25).

Table 1
ASCII files used for reclassifications and color assignments.

ASPECTREMAP	25COLORREMAP	RGBTABLE
lowest-output 1	19 : 1	19 153 153 153
0 22 : 1	21 : 2	21 147 166 89
22 67 : 2	22 : 3	22 102 153 102
67 112 : 3	23 : 4	23 102 153 136
112 157 : 4	24 : 5	24 89 89 166
157 202 : 5	25 : 6	25 128 108 147
202 247 : 6	26 : 7	26 166 89 89
247 292 : 7	27 : 8	27 166 134 89
292 337 : 8	28 : 9	28 166 166 89
337 359 : 1	31 : 10	31 172 217 38
	32 : 11	32 77 179 77
	33 : 12	33 73 182 146
	34 : 13	34 51 51 204
SLOPEREMAP	35 : 14	35 128 89 166
lowest-output 10	36 : 15	36 217 38 38
0 5 : 10	37 : 16	37 217 142 38
5 20 : 20	38 : 17	38 217 217 38
20 40 : 30	41 : 18	41 191 255 0
40 220 : 40	42 : 19	42 51 204 51
	43 : 20	43 51 204 153
	44 : 21	44 26 26 230
	45 : 22	45 128 51 204
PLUSREMAP	46 : 23	46 255 0 0
lowest-output 19	47 : 24	47 255 149 0
0 19 : 19	48 : 25	48 255 255 0

RGBTABLE lists preliminary colors in use before the map was exported to the Macintosh environment for color-scheme work in HVC (see Appendix for final color specifications).

SHADEEDIT was used to specify colors from HLS for each of the 25 values (25COLORS.SHD). Display of the combined slope and aspect grids, using the new remap table and new SHADESET, was performed with the following commands:

```
GRID: AP SHADESET 25COLORS.SHD
GRID: GRIDSHADES HVPLUS2 VALUE 25COLORREMAP NOWRAP
```

ARC's GRIDIMAGE command was used to convert the gridded data to a TIF file format that was exported to other graphics environments. GRIDIMAGE, however, required a color-map table rather than a SHADESET (it did not accept HLS color specifications). Thus, an ASCII table (RGBTABLE, Table 1) of red-green-blue equivalents to the HLS specifications was created to produce TIF output (the RGB values were recorded earlier using on-screen translation of HLS values within SHADEEDIT).

```
ARC: GRIDIMAGE HVPLUS2 RGBTABLE HVPLUS2.TIF TIFF NONE
```

The TIF file was moved from the Sun to a Macintosh environment using network file transfer protocol (FTP). It was then opened in Adobe PhotoShop and final color manipulation with HVC and graphic editing were performed.

SUMMARY

Systematic variations in all of the perceptual dimensions of color were used for the aspect/slope color scheme discussed in this paper. Eight aspect categories were mapped with an orderly progression of colors around the hue circle. Three slope categories were mapped with differences in saturation, and near-flat slopes were mapped with gray for all aspects. Lightness sequences were built into both the aspect and slope progressions to produce systematic perceptual progressions of color throughout the scheme (among all 25

map categories). These lightness sequences approximated relief shading and were crucial for visualization of the shape of a continuous terrain surface. Use of the HVC color system allowed easy specification of 25 colors that accurately met these interlinked color-appearance criteria. Our aspect/slope color scheme allows accurate terrain visualization while simultaneously categorizing the surface into explicit aspect and slope classes represented within a single, two-dimensional, planimetric map.

REFERENCES

Chang, K., and B. Tsai, 1991, The Effect of DEM Resolution on Slope and Aspect Mapping, *Cartography and Geographic Information Systems*, 18(1):69-77.

GIS World, 1991, MKS-ASPECT Enhances Color Surface Renderings, *GIS World*, 4(October):30-32.

Moellering, H., and A.J. Kimerling, 1990, A New Digital Slope-Aspect Display Process, *Cartography and Geographic Information Systems*, 17(2):151-159.

Taylor, J., A. Tabayoyon, and J. Rowell, 1991, Device-Independent Color Matching You Can Buy Now, *Information Display*, 4&5:20-22, 49.

ACKNOWLEDGEMENTS

The data, software, and hardware used in development of the aspect/slope map scheme were provided by the Steven and Mary Birch Foundation Center for Earth Systems Analysis Research (CESAR) in the Department of Geography at San Diego State University. We appreciate the skill and assistance of Dave McKinsey, the technical manager of CESAR.

APPENDIX

Specifications of final colors for aspect and slope categories. CIE 1931 **x**,**y** chromaticity and luminance (**Y** in cd/m^2) were measured with a Minolta Chroma Meter CS-100 from a 19-inch SuperMac monitor calibrated (with a SuperMatch system) to approximate a generic Apple 13" color monitor with a D65 white point and gamma of 1.4. Please be aware that use of the **RGB** values will not reproduce the aspect/slope color scheme if your monitor is much different than ours.

aspect	color	H	V	C	Y	x	y	R	G	B
Maximum-slope classes (greater than 40 percent slope)										
SE	P	285	35	51	10.3	.240	.135	108	0	163
S	M	325	50	60	19.6	.350	.198	202	0	156
SW	R	5	65	63	33.7	.442	.319	255	85	104
W	O	45	80	48	53.8	.421	.421	255	171	71
NW	Y	85	95	58	81.1	.395	.512	244	250	0
N	GY	110	80	56	55.3	.339	.546	132	214	0
NE	G	135	65	48	35.2	.252	.486	0	171	68
E	B	245	50	46	19.8	.180	.191	0	104	192
Moderate slopes (20 to 40 percent slope)										
SE	P	285	47	34	17.4	.262	.209	119	71	157
S	M	325	57	40	25.1	.332	.240	192	77	156
SW	R	5	67	42	34.3	.390	.316	231	111	122
W	O	45	77	32	47.3	.377	.384	226	166	108
NW	Y	85	87	39	63.7	.359	.445	214	219	94
N	GY	110	77	37	50.6	.319	.451	141	196	88
NE	G	145	67	33	37.0	.253	.400	61	171	113
E	B	245	57	31	25.7	.221	.238	80	120	182
Low slopes (5 to 20 percent slope)										
SE	p	285	60	17	27.5	.280	.272	140	117	160
S	m	325	65	20	32.6	.311	.282	180	123	161
SW	r	5	70	21	37.2	.339	.317	203	139	143
W	o	45	75	16	41.8	.328	.348	197	165	138
NW	y	85	80	19	49.7	.321	.375	189	191	137
N	g y	110	75	19	45.4	.305	.381	152	181	129
NE	g	155	70	16	39.4	.270	.348	114	168	144
E	b	245	65	15	32.7	.260	.284	124	142	173
Near-flat slopes (0 to 5 percent slope)										
none	gray	0	72	0	39	.290	.317	161	161	161

THREE-DIMENSIONAL (3D) MODELLING IN A GEOGRAPHICAL DATABASE

Béatrix de Cambray

Laboratoire MASI (CNRS, Universités de Paris VI et de Versailles-St Quentin)
45 avenue des Etats-Unis, F-78000 Versailles, France.
Fax: 33 1 39 25 40 57 - E-mail: Beatrix.De-cambray@masi.uvsq.fr

ABSTRACT

While most current Geographical Information Systems (GIS) are 2D GIS, more and more application domains as geology, civil and military engineering, etc, need 3D GIS. The particularities and capabilities of a 3D GIS are outlined in this paper. Although 3D GIS are not much developped, some studies have already been done, especially in order to represent 3D data for geological or architectural applications. Such 3D models are generally based upon Computer-Aided Design (CAD) models. The use of 3D CAD models for representing 3D in GIS is discussed in the paper. As 3D GIS represent a 3D map, not only the surface of the terrain but also the entities set on the relief, e.g. buildings, have to be represented. The proposed 3D model for a GIS consists of representing 3D entities set on a DTM. These entities are represented with different approximation levels, the first two being used as index and processes accelerators and the last offering an efficient visualization support. The first two levels are entities decomposition into minimal bounding rectangular parallelepipeds (MBRP). The last level is the external representation: it is a Boundary Representation (BR). Indexation is shortly discussed in the paper. An existing geographical database system is currently extended to support these 3D entities.

1. INTRODUCTION

Most current geographical database systems are restricted to 2D information. However, the third dimension appears to be more and more necessary for many application domains as geology, civil and military engineering, town planning, robotics, etc... These applications require not only the modelling and the visualization of 3D data but also the manipulation of these data. For example, in a town planning project, an impact study needs examining 3D visual intrusion caused by a building on the terrain and needs being able to move the building location. It is comparable with the design of a 3D scale model which would furthermore be easy to modify.

3D data involved in a GIS may be subsoil data or relief data or 3D geographical data that can either be human made entities like bridges or naturally occuring entities like forests. A 3D entity is a volumetric object which can be convex or concave and which may contain holes... A cube and a sphere are examples of 3D entities.

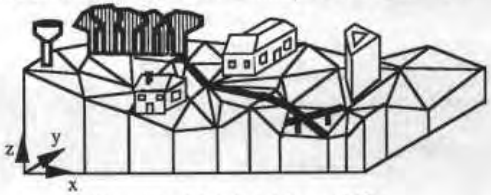


Figure 1: 3D Model for a GIS

Three-Dimensional GIS represent a 3D map. Consequently it is interesting to distinguish between the surface of the terrain and the entities set on the relief, e.g. buildings, bridges, forests, etc... In the proposed model, 3D geographical entities are set on a Digital Terrain Model (DTM) which describes the relief (See Fig. 1). As discussed later, a DTM alone is not sufficient to represent a 3D map.

It would also be possible to represent geological data by assuming geological layers are 3D entities limited by surfaces like DTM. A geological layer is then a polyhedron (the

slice between DTM). In this paper we limit our scope on over the ground representation.

One the one hand, modelling the third dimension in a GIS provides the following main advantages. 3D is necessary to fully represent spatial reality and to remove some ambiguities, e.g. a road hanging above another one or a roman aqueduct with several levels. Those Geographical Information Systems (GIS) restricted to 2D usually rely on a special symbolization to represent a segment corresponding to two overlapping roads; furthermore, the user have to clearly indicate which road is above the other while this information is implicitly captured by the internal model in a 3D GIS. The elevation is lost in a 2D GIS while it may be important. 3D purpose is to represent as accurately as possible any three-dimensional shapes.

On the other hand, a 3D representation have the three following important disadvantages: (1) the volume of storage is generally very important, (2) geometrical computations are more complex with a 3D than with a 2D representation, and (3) 3D data capture is rather difficult. In particular, uncertainty about 3D data (as relief or subsoil data) is greater than about 2D data.

The paper is organized as follows: intermediate solutions between 2D and 3D are first reviewed, then the particularities and capabilities of a 3D GIS are presented. An overview of 3D CAD models is given in section 4. Our solution is then described. Section 8 corresponds to a short discussion on index. At last, we conclude.

2. 2.5D AND DTM

2.1 Digital Terrain Model (DTM) and 2.5D

Spatial data are 3D data. As a result, elevations are lost in a 2D map. Several intermediate solutions between 2D and 3D, as DTM and 2.5D may be defined. As many others, we define here 2.5D as a representation mode of a flat (or nearly flat) entity such that z= f(x, y) and f is a true function. This definition encompasses both usual 2.5D maps and DTM.

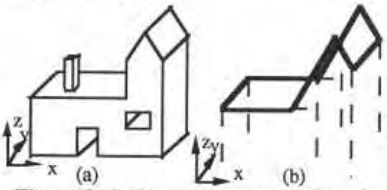


Figure 2: 2.5D representation (Fig. 2 (b) is a 2.5D representation of the entity plotted in (a))

With usual 2.5D maps, each geographical feature (or part of) such as a forest, a field or a road becomes an individual entity represented by its 2D-boundary augmented with a z-coordinate. While linear entities are well approximated, inner surface elevation variations are not captured. A non homogeneous approximation of the underlying terrain is provided whose accuracy depends on the local entity density.

With DTM the terrain itself is a single entity represented by a sample of points corresponding to a facetized surface with these points as nodes. Hence, a tesselation covering the whole terrain is achieved. Different tesselations (regular, irregular or hybrid) may be built with the same dataset. Usually an height is associed with each node of the tesselation. Other informations like the slope may be associated with a face of the mesh. Hence, we have a representation of the terrain offering an approximation of the heights. DTM extensions have been proposed where the function z= f(x, y) is assumed continuous and sufficiently differentiated: an example is given by spline function. However, in DTM specific entity boundaries are usually lost. A common solution to this

problem is the combination of both a DTM and a 2.5D representation for a single map. (Miller 1958, Ebner 1992, Fritsch 1992)

Keeping "flat" entities in mind, another solution for 2.5D is possible: instead of augmenting a boundary-based 2D representation of an entity with elevation values, a 2.5D representation could be derived from the 3D representation of an entity given by its Boundary Representation (BR). Only upper facets of the BR are kept (as expected, we define upper facets as those visible - vertically - from above). Fig. 3 (b) is the 2.5D representation of the entity of Fig. 3 (a). Such a representation can be extended at low cost to capture the vertical thickness of entities such as most buildings.

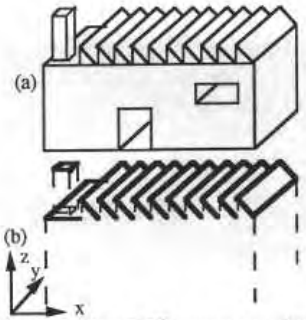


Figure 3: Our 2.5D representation

2.2 3D Versus 2.5D

A 2.5D GIS is able to represent two overlapping roads (assuming that a road has no thickness) but not an arch, nor a roman aqueduct with several levels (e.g. Fig. 4) because it can not represent entity thickness. In such a case, the user does not know if it is feasible to get between the columns of the arch. 2.5D models only give a rough approximation of an entity which may appear to be insufficient for applications like navigation. Likewise they can not represent holes in an entity. Complex 3D entities (e.g. Fig. 4), especially those with holes, openings, etc..., can only be represented with full 3D because of the need for an arbitrary number of elevations for a given (x, y) position.

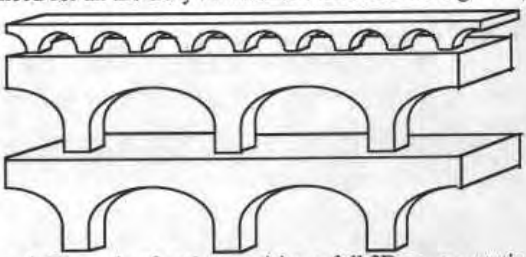


Figure 4: Example of entity requiring a full 3D representation

As representing a 3D map takes more than the representation of the surface of the earth and thus more than one z-coordinate for a given (x, y) position, a DTM or a 2.5D model are not sufficient. Reversely, as a full 3D representation of the terrain proves too costly and awkward to use, we have to model entities set on a representation of the surface of the earth given by a DTM.

3. 3D GIS

Although 3D GIS are not much developped, some studies on 3D GIS have already been done (Jones 1989, Raper 1989, Pigot 1991 and 1992), especially in order to represent 3D data for geological or architectural applications. Such 3D models are generally based upon Computer-Aided Design (CAD) models as the voxel model.

3.1 The Particularities of a 3D GIS

3D requirements of GIS differ from those of CAD systems on at least three accounts:

entity "flatness", data acquisition mode, and geographical entity shape.

Entity "flatness" arises because the data size in GIS is not the same as in CAD or in architecture in the sense that geographical data are drastically more expanded among the plane (e.g. 100 km x 100 km) than among the third dimension (the highest point in the world is the Everest which culminate at 8882 m above the sea level, and the highest mountain in the world is Mauna Kea (Hawaï) that reach 10203 m 4205 m of which are under the sea. In the case of buildings, their height do not top a few hundred meters).

In GIS, the data acquisition mode implies a descriptive approach; this is true not only for naturally occuring entities as forests and mountains but also for superstructures as buildings or bridges that can not be easily decomposed into simple (volumetric) primitives. Note that this is the same problem as trying to rebuild the CSG representation from a descriptive CAD representation such as voxel or octree.

Natural entities that are represented in a GIS (e.g. forest) seldom have a regular shape contrary to human made entities.

What is more, a 3D GIS does not deal uniquely with 3D volumes. The terrain itself is merely a 3D surface: unless the surface of the terrain is fictitiously closed by a "floor" and by upright "walls" joining the surface and the floor or is roughly approximated by contiguous parallelepipeds with differents heights, the relief can not be seen neither as a 3D closed entity, nor as a solid (in the sense of "solid" defined in (Requicha 1980)). The above-mentioned CAD models which are models for solids can not exactly represent the relief: a DTM is more appropriate for this purpose.

3.2 The Fundamental Capabilities of a 3D GIS

The operations of a 3D GIS can be sorted into four basic classes (Bak 1989):

(1) entity definition and modification - Boolean set operations (union, intersection and difference) and the geometric transformations (rotation, translation, scaling) have to be easily computed;

(2) visualization - which must be efficient;

(3) integral property computation - volume, surface area, distance or mass (it is specially important for geological applications);

(4) information retrieval - The system has to be able to do spatial and non-spatial queries and must therefore provide an efficient clustering method, an index in order to speed up searches.

4. 3D GIS VERSUS 3D CAD MODELS

4.1 An Overview of Main 3D CAD Models

CAD models can be sorted in two groups: volume representation and boundary representation (Foley 1990).

In the first group, an object is described as a combination of volume primitives (e.g. a cube). This group consists of the decomposition models which comes in two classes: either the model is a constructive one, either it is a descriptive one. Constructive Solid Geometry (CSG) and Primitive Instancing (PI) belong to the first class. In the second class there are the octrees, polytrees and their extensions and also the voxel model. As it is outlined by the name of this class, a constructive model is a model where an object is built by assembling components. On the contrary, a descriptive model do not aim to decompose the objects into its constituant parts but it decomposes the object space into a collection of contiguous nonintersecting solids; these solids are primitives (e.g. cubes)

but not necessarily of the same type as the original object. They may not be object constituants. They are either partially or entirely occupied by the original object or empty.

In CSG, solids are built combining elementaries shapes as cylinders, cubes or spheres, by means of regularized Boolean set operators. Primitive Instancing (PI) differs from CSG in that PI primitives are only 3D shapes that are relevant to the application domain and are not limited to certain simple 3D solids. PI is an adequate representation if the set of entities that have to be modelled is well known and does not vary. It is often used for relatively complex entities, such as gears, that are tedious to define with a CSG model, yet are easily described by a few high-level parameters. If the entities to be represented are well adapted to the model, CSG and PI provide concise representations.

Cell-decomposition is an irregular decomposition of an entity into cells (e.g. a cube, a tetrahedron) and a representation of each cell in the decomposition. The voxel model (or spatial occupancy enumeration) is a special case of cell decomposition in which the solid is decomposed into identical cells arranged in a fixed grid.

In an octree (or its extensions: polytree, ...), the whole space is recursively divided into cubes giving a hierarchical structure. Leaves nodes are an accurate to a given degree cubical approximation of the entities constituents.

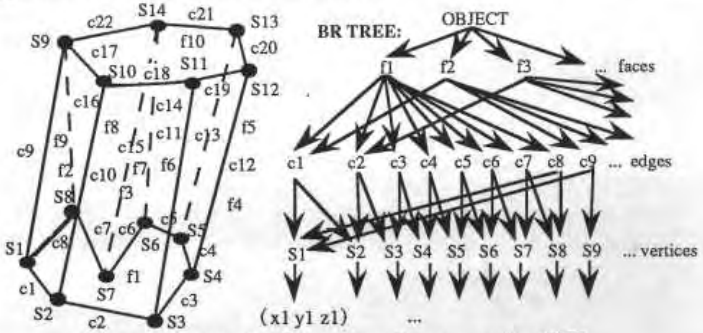


Figure 5: Example of Boundary Representation (BR)

In the second group, a solid is described by its bounding surface. These models are based on surfaces and have to include enough information in order to know how the surfaces join for defining closed volumes. Boundary Representation (BR) belongs to this group. BR represents a solid in terms of its bounding surfaces: solids are represented as union of faces, each of which is defined by its boundaries ie the vertices and edges bounding it. The model primitives are therefore: faces, edges and vertices. Every entity is represented by a constant depth tree. Such an entity representation is a polyhedral one because faces are planes and edges are straight lines: it can only be an approximation for non polyhedral entities. Fig. 5 illustrates a BR.

The most common CAD models are CSG and BR. For a complete survey of 3D modelling, we refer the reader to (Foley 1990) or (Cambray 1992).

4.2 Discussion about the Use of CAD Models for 3D GIS

In cartographic applications, the decomposition of an entity into faces, edges and vertices is a straight extension of the 2D case: BR seems natural. Unlike volume representation, BR model is easy to manipulate for the visualization but is restricted to polyhedral representations. BR is usually neither concise, nor efficient for computing boolean set operations or integral properties or for models validations.

CSG is a constructive model better fitted to the representation of entities built by men (by assembling simple constituents (sphere, cube, ...)), such as plumbing parts. Hence, a constructive (volume) representation such as CSG has to be eliminated.

The voxel model and spatial decomposition models are often used for geological applications. Such representations provide an easy way for computing integral properties or location problems or Boolean set operations. But these representations tend to be extensive on storage space and inexact for the purposes of representing entity shapes, particularly if they have curved surfaces.

More generally, integral properties or Boolean operations calculations can be easily achieved using a volume representation. Therefore, any storage-efficient descriptive (volume) representation can be selected. Storage efficiency disqualifies voxel mode. Note also that bounding volumes may be used for applications looking at entities as obstacles. Different representations can be chosen.

Unfortunately, there exist no CAD model perfectly adapted to 3D modelling in GIS.

5. A 3D MODEL FOR A GIS

In order to overcome CAD models shortcomings, a new solution (for representing 3D geographical entities) is described. In the proposed model, 3D geographical entities are set on a Digital Terrain Model (DTM). Every entity is represented with a multiresolution 3D model, i.e. having several approximation levels.

5.1 Multiresolution Model for Representing a Geographical Entity

The most important functions of a 3D GIS differ on their representation requirements. With a single representation mode a system can not be optimal for all applications. On the contrary, a multiresolution mode combines the advantages of several representations while limiting their disadvantages. To offer different approximation levels comes to the same thing as using for computing a given operation (e.g. indexation, Boolean operations) the most adapted representation for this operation.

Using different approximation levels provide for the selection of the appropriate level according to a given scale or priority in some computation. These levels are also used by the system for making rough approximations before the actual accurate, but expensive, computation is made.

It should be possible to gather together the same level representations of several nearby (semantically and spatially) entities: for example to gather together houses belonging to the same block. This operation is nicknamed "semantical zoom" in the sense that we obtain an increase effect: an entity (block) becomes several entities (houses), this gathering can be made because it is meaningfull: a block is a group of houses.

There are three approximation levels for each entity. The proposed model is an internal model. The external model is a set of faces. Indeed, the user provides a points set which gives a set of faces and allows then to compute the entity BR. We assume therefore that the BR is the most accurate entity representation. The system immediately computes the other levels from the provided BR. The user can only modify the BR, the modifications are then done on the other levels.

6. THE DIFFERENT APPROXIMATION LEVELS

6.1 The First Approximation Level

The highest level is the minimal bounding rectangular parallelepiped (MBRP) of the 3D entity shape. The MBRP is parallel to the axis of the referencing system of the world in order to ease the computations.

The MBRP of the geographical entity plotted in Fig. 6 (coloured with grey in Fig. 7) is shown in Fig. 7.

This level can be used as an index key because it is a bounding volume of the 3D entity which is the extension of the bounding rectangle in 2D GIS. This bounding volume allows an easy entity localization. It is then possible to cluster thanks to this approximated shape. This bounding volume may be also useful to speed up 3D entities operations (e.g. intersection): it allows a rough but quick approximation of the operation result. Indeed, it is much less expensive to compute Boolean operations with only one parallel to the axis MBRP than with any other representations (CSG, BR, octree,...). On the contrary, the result is only an approximation. As a matter of fact this level realize a filter that eliminates non adequate answers. For example, if the MBRP of two different entities do not intersect, then the two entities do not intersect: the answer is therefore given avoiding the intersection computation on two entity representations which may be extremely complex.

As far as semantical zoom is concerned, it is easy to calculate the MBRP of a group of entities semantically and spatially closely related. This allows a much more speeding up of spatial predicates.

Figure 6: Example of a 3D geographical entity

Figure 7: the first approximation level

This level is therefore useful for the system as an index and as processes accelerator, especially spatial predicates. The integral properties computation is very fast but the result is a very rough approximation.

6.2 The Second Approximation Level

The second level is an entity subdivision in at most n MBRP (n limited by system, e.g. n= 8). The MBRP are parallel to the axis of the referencing system of the world. Fig. 8 describes the eight MBRP (grey or shaded) of the second level decomposition of the entity plotted in Fig. 6.

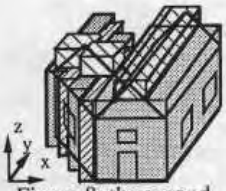


Figure 8: the second approximation level

The way chosen to calculate this representation level is equivalent to the octree construction. Indeed, the MBRP of the first level is recursively divided into eight parallelepipeds, then this decomposition is enhanced.

This level provides a better approximation, not only of the processes but also, of an entity shape (which is usually sufficient for many applications). Computations remain fast. n is fixed by the system to ease computation and storage. The objective of this level is to speed up Boolean operations and integral properties computations while providing a relatively good approximation of the result. It realizes a more accurate filter than the one of the previous level because on the one hand, the global shape of the entity is better approximated than previously and, on the other hand, it is feasible to know which ones among the n parallelepipeds have, for example, an empty (resp. non empty) intersection with the parallelepipeds of an other entity: to enhance the response, it will only be

necessary to consider the part of representation fitted with these parallelepipeds. Integral properties computations will be much better than with the first level.

This level is therefore useful for the system not only to speed up processes but also to quickly retrieve an entity part.

6.3 Remark about the First Levels

The first two levels may appear to be very much like a CSG representation but restricted to only one primitive (the parallelepiped) and only one operation: the gluing which is a restricted form of union. However, the parallepipeds considered in this model are not entity constituents: they are bounding parallelepipeds. The representations are therefore "true" in the sense that they include the entity. The operations on these representations provide approximations because a bounding parallelepiped is usually not fully filled by the entity. These operations do not generate errors because there is no part of the entity outside of the union of MBRP. The use of bounding volumes as a test to speed up processes (e.g. ray tracing) is a usual image synthesis method.

It would be possible to allow the user to choose the decomposition level. For instance, if the user wants three recursive decompositions of an entity, the three levels of the octree are computed for this entity; then, the decomposition is enhanced. This would be primarily used for visualization. Note that the corresponding accuracy may be as good as a user wishes to the price of additional complexity.

6.4 The Third Approximation Level

An additional third level is provided to ease the connection between our system and CAD softwares. This level is a BR because BR is the most common CAD model with CSG. As mentioned earlier BR has been prefered to CSG.

The chosen BR is a non manifold one in order to be able to uniformly consider 3D, 2D or 1D entities or 3D, 2D and 1D hybrid entities (Masuda 92). Such geographical entities do satisfy a generalization of the Euler-Poincaré formula (but not necessarily the usual one): they may have dangling faces or dangling edges.

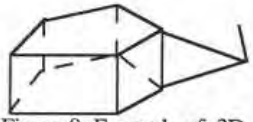


Figure 9: Example of 3D, 2D and 1D hybrid entity

This latter level provides almost perfect rendering and great accuracy in geometric computations. The entity BR is a good representation even if the entity is a polyhedral one. If we do not want to be restricted to polyhedra, we must consider a BR generalized to curves and curved surfaces.

7. CONCLUSIONS ABOUT THE MODEL

The main advantages of our model are the following ones:

- our model provides a spatial index (with the first level);
- operations are speeded up. As BR is the more accurate provided representation, the accurate computations have to be done with the BR. It is better to compute operations with the first two levels representations and particularly with the second one before the actual accurate, but expensive, computation is made (if the second level filter has not eliminated the entity). The benefits of computing with entity approximation not only lie in the fact that the system can filter the answers but also in the fact that it can straightly obtain, from the second level representation, the part of BR associated with one of the n parallelepipeds. Indeed, as far as the Boolean operations are concerned, the second level computation allows the system to determine what MBRP are really important for the

computation, then the system only need to make the computation with the part(s) of BR associated to these parallelepiped(s). This enables the system to compute with only part of the BR. The operation complexity is then reduced. The two first level therefore allows efficient Boolean operations or integral properties computations;

• our model provides an efficient visualization thanks to BR level.

However, there are some disadvantages:

• accurate computations are expensive as BR is not efficient for Boolean or integral properties calculations. This disadvantage is reduced by the use of the speeding up techniques previously presented;

• entity shape may be roughly approximated by the first two levels as these levels provide a polyhedral representation and as the MBRP are parallel to the axis of the referencing system of the world and not of the entity. But, for the scale considered in a 3D GIS, a very accurate representation is not always necessary and is not realistic from a data capture point of view;

• BR is generally expensive from the storage volume point of view but the BR level is essential in the sense that the data provided to the system are set of faces (i.e. a representation mode close to BR). Furthermore, BR allows a relatively accurate representation of entities. On the other hand, the storage volume of the first two levels (n° 1 and n° 2) is limited and small.

8. INDEX

As we have already shown, spatial extent is larger among the plane than among the third dimension. Therefore, a 2D spatial index is sufficient for a 3D GIS. However, the first levels provide a 3D spatial index. Indeed, the MBRP of the first level provides a 2D spatial index (we only consider xmin, xmax, ymin, and ymax so that we have a bounding rectangle) and a rough elevation index (with zmin, zmax) which can ease the selection on the z axis. This elevation index is sufficient in the sense that, in the geographical world, (1) the spatial extent among the z axis is not very large, and (2) the number of objects which have the same planar location is rather small (a counter-example is the case of geological applications where there are many geological layers). The second approximation level allows to select a part of the 3D entity.

9. CONSISTENCY BETWEEN 2D AND 3D CARTOGRAPHICAL MODELS

Our model is compatible with 2D cartographic models. With 3D data, users can only consider 2D maps through a plane projection of a 3D map. It is therefore necessary for a 3D GIS to be able to display, manipulate and query a 2D map and to obtain again the 3D map as soon as it is needed.

A user may display and manipulate a 2D rendering of some map while keeping 3D correctness: for example, if there is visually an intersection between a river and a bridge on a 2D projection, the system properly answers that they do not intersect. A 2D intersection may correspond to a 3D null-intersection (i.e. a crossing: one entity is above the other). To this end, the process has to consider 3D data, even if a query is done on the 2D map.

Some 3D operators may be seen as refinement of 2D operators. Hence, the 3D extension of 2D intersection operator (cf. Fig. 10) is either a (true) 3D intersection, or a crossing.

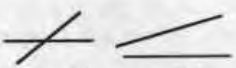


figure 10: intersection

10. CONCLUSION

The particularities of a 3D GIS lies in the following points: entity "flatness", data acquisition mode, geographical entity shape, and the nature of data involved (relief, subsoil and 3D geographical data). Their capabilities are the definition and the modification of entities, visualization, computations, information retrieval. So that a 3D GIS has to meet these requirements.

This leads us to propose a model for 3D GIS which consists of representing 3D entities set on a DTM. These entities are represented with different approximation levels, the first two being used as index and processes accelerators and the last offering an efficient visualization support. The first two levels are volume representations: they are entities decompositions into MBRP. The last one is a surface representation: it is a BR. It is the external representation.

An existing database system based on an extended relational model and supporting built-in spatial data types, operations, and indexes, GéoSabrina, is currently extended to support these 3D geographical entities. GéoSabrina, which is developped at the MASI Laboratory, uniformly manages both semantical data and cartographical data (location and geometric shape) (Larue 93). New spatial data types are defined along with operations upon those types. Hence, two additional data types are defined: one for the pure 3D entities (ie 3D entities with no dangling faces, dangling edges nor isolated points) and the other for hybrid entities. Examples of operations upon these entities or/and the DTM are union and geometrical projection.

This model is also used to solve spatio-temporal problems (Yeh 93).

REFERENCES

Bak, P. R. G., and Mill, A. J. B. (1989), *Three dimensional representation in a Geoscientific resource management system for the minerals industry*, In: Three dimensional applications in Geographical Information Systems, Jonathan Raper (ed.), pp. 155-182. Publisher: Taylor & Francis.

de Cambray, B. (1992), *Modélisation 3D: état de l'art*, Technical report, MASI 92.71. Publishers: Institut Blaise Pascal & Laboratoire MASI, Paris.

Ebner, H. and Eder, K. (1992), *State-of-art in digital terrain modelling*, In: Proceedings of EGIS'92, Munich, pp. 681-690.

Foley, J. D., van Dam, A., Feiner, S., and Hughes, J. (1990), *Computer graphics, principles and practice*. Publisher: Addison-Wesley Systems Programming Series.

Fritsch, D. and Pfannenstein, A. (1992), *Conceptual models for efficient DTM integration into GIS*, In: Proceedings of EGIS'92, pp. 701-710.

Jones, C. B. (1989), *Data structures for three-dimensional spatial information systems in geology*, International Journal of Geographical Information Systems, 3 (1), pp. 15-31.

Larue, T., Pastre, D., and Viémont, Y. (1992), *Strong Integration of Spatial Domains and Operators in a Relational Database System*, SSD'93, LCNS n° 692, pp. 53-72.

Masuda, H. (1992), *Recent progress in 3D geometric modeling*, MICAD 92, Tutorial, Paris.

Miller, C. L. and Laflamme, R. A. (1958), *The Digital Terrain Model - Theory & Application*, Photogrammetric Engineering, XXIV (3), pp. 433-442.

Pigot, S. (1991), *Topological models for 3D spatial information systems*, In: Technical papers, ACSM-ASPRS Annual Convention, Baltimore, Auto-Carto 10, 6, pp. 368-392.

Pigot, S. (1992), *A topological model for 3D spatial information system*, In: P. Bresnahan, E. Corwin, and D. Cowen (ed.), Spatial Data Handling'92, Charleston, USA, 1, pp. 344-360.

Raper, J. and al. (1989), *Three dimensional applications in Geographical Information Systems*, Jonathan Raper (ed.). Publisher: Taylor & Francis.

Requicha, A. A. G. (1980), *Representations for rigid solids: theory, methods, and systems*, Computing Surveys, 12 (4), pp. 437-464.

Yeh T.S. and de Cambray B. (1993), *Time as a geometric dimension for modelling the evolution of entities: a 3D approach*, Second International Conference on Integrating Geographic Information Systems and Environmental Modelling, Breckenbridge, Colorado, USA, September 1993.

HOW MULTIMEDIA AND HYPERMEDIA ARE CHANGING THE LOOK OF MAPS

Sona Karentz Andrews and David W. Tilton
Department of Geography
University of Wisconsin - Milwaukee
P.O. Box 413
Milwaukee, Wisconsin 53201

ABSTRACT

The ideas of what constitute the presentation format of maps have changed. We now have the opportunity to create not only maps printed on paper or displayed on a computer monitor from a digital file, but we can present them on a television screen from a digital Photo CD, an analog laserdisc, or videotape. We can even add motion, sound, and interaction to the cartographic image. This great diversity in types of media and the opportunity to integrate and link these various forms to cartographic information is changing the way we create and use maps. As cartographer's begin to explore and incorporate multimedia and hypermedia technology into their work they are confronted with a host of tools and ideas. This paper discusses a range of multimedia hardware and software available for use in cartographic displays and presents the results of a research and demonstration grant funded by the United States Department of Education to create an interactive videodisc on mapping. The videodisc contains thousands of still frame images of over 600 maps and 25 minutes of motion video, all linked to a digital database containing information about each map. Access to the map images and database are possible through an interface designed specifically for this project.

INTRODUCTION

In the not-so-distant past the sole medium for which we designed and created maps was print-on-paper. Whether the map was to be published as a single sheet; bound in an atlas, book, or journal; or destined to remain in manuscript form, the stages of map compilation, development, production, and the final product were essentially paper-based. In the 1960s automated cartography gave us a new set of tools for map creation. Cartographers were able to issue sets of commands and coordinates via a terminal or punch cards to generate output (in the form of a paper map). Recent technological advancements have created a computer desktop which has, for the most part, replaced hand-drawn, pen and ink, drafting table map production. By and large, however, the output and final form of our maps is still paper.

Multimedia has the potential to change all that. It has already made tremendous transformations in the how graphics are presented and communicated, and maps will be no exception. One only needs to pick up the latest issue of any computer trade magazine to know that motion, on-screen

presentations, sound, and color have become integral components in graphic communication.

Cartographers cannot and should not shelter maps from the changes that are taking place in graphic information capture, creation, manipulation, and presentation. We have, in fact, already started to make use of some of these tools to improve and facilitate the processes we use to create maps through mapping and illustration software and high-end image output devices. These technologies, however, offer even greater possibilities to alter cartographic practices. Not only can they be adapted to facilitate and improve traditional map creation methods, they have the potential to change the look of maps and how we communicate spatial information. It is worth restating this idea to make it clear that what is being discussed here are not merely changes in map compilation and production methods, but how these new technologies will change how maps are conceived, how they communicate spatial information, and how they will be used. The incorporation of map animations in recent electronic atlases and encyclopedias are just a few examples of cartographic ventures into this new territory.

As cartographers begin to explore and incorporate multimedia (and hypermedia) technology into their work they will be confronted with a barrage of technical information and new terminology. This paper discusses some of the multimedia and hypermedia tools that have potential for use in cartographic displays. Our intent is not to provide a comprehensive list of all the software, formats, and hardware available. Not only would such a list be enormous (if it were even possible to know the extant list of items), but it would certainly be out-of-date even before the words are entered into this document. Given the dynamic nature of multimedia and hypermedia technology, our primary goal is to present readers of this paper with a quasi-glossary of multimedia and hypermedia terms that cartographers are likely to confront in creating and displaying maps. Along with these terms we have provided descriptions of some specific items. The majority of information is applicable to the Macintosh platform, since this is the one we have the most experience with. We hope that this organizational framework and treatment provides a concise and useful way to inform cartographers of the availability and use of some tools and techniques. We caution the reader to keep in mind the "packaging date" of this information and the limited shelf-life that some of these items may have. We conclude this article with a description and discussion of a multimedia project aimed at providing hypermedia access and display of different types of maps.

MULTIMEDIA GLOSSARY

Multimedia

A 1992 issue of *PC Publishing and Presentation* states "in an industry rife with buzz-words, multimedia has become the buzziest of them all." Most simply stated, multimedia is the combined use of various media such as text, graphics, photos, video, and sound in computer presentations and/or stand alone applications.

Hypermedia

Hypermedia is a form of multimedia presentation in which associations between pieces of information or ideas are identified by predefined, user-activated links. The user activates these links by interacting with the computer

through on-screen objects such as buttons. Relationships between the various pieces of information in a hypermedia application differ from those contained in a traditional database in that hypermedia links do not identify every instance that fits a specified criteria (as in a database search), but instead identify links between the current ideas or pieces of information and some other in the application.

Animation

Animation is based on the phenomena known as the persistence of vision in which the mind retains an image for a split second after it is no longer visible to the eye. If the first image is replaced by a second relatively similar image, the illusion of motion is created. Between 16 to 24 images per second are required to create the illusion of fluid motion. Computer animation programs are not only designed to play the images (frames) that produce motion, but also to reduce the effort in creating those frames. Some programs (such as Macromind Director) are based on traditional techniques such as *cel* animation where key frames (ones that define the range of a specific motion) are created, and then *inbetween* frames are generated by the computer. Others programs (such as Swivel 3D) work with defined three dimensional objects that are then moved. Complex animations require large amounts of storage space, a relatively fast CPU and hard drive, and often an accelerated monitor. Most animation programs on the Macintosh can export files in the PICS format, or as QuickTime movies. As QuickTime movies they can be imported and played by a large number of programs including word processors.

Authoring Tools

Authoring tools are programs designed to assist in creating multimedia or hypermedia applications that range from relatively simple "slide-show" presentations (Aldus Persuasion) to sophisticated interfaces for complex interactive applications (Macromind Director). Although the distinction between authoring and more general programming tools is not entirely clear, authoring tools are optimized for linking objects of differing media (text, graphics, sound, photographs, video) into some sort of directed sequence or presentation. The more powerful authoring tools also provide predefined controls for external devices (such as videodiscs, compact discs, and videocassette recorders). The most sophisticated authoring tools allow the author to write scripts that control the course of the interaction. Scripting languages are generally based on HyperTalk, a high-level programming language that fully implements object-oriented programming principles. The majority of authoring tools that provide scripting environments also allow the scripts to call other programs external to the application itself. Examples of authoring tools include Action (Macintosh), Authorware Professional (Windows and Macintosh), HyperCard (Macintosh), Macromind Director (Macintosh), Passport Producer (Macintosh), SuperCard (Macintosh and Windows), and Toolbook (Windows).

Authorware Professional. Authorware began as an authoring program oriented toward computer-based training. It utilizes icons that are programmed to perform predetermined functions such as animating objects, displaying graphics, pausing the flow of the program, creating decision paths, establishing interactions that wait for user responses, and performing calculations. These icons provide the ability to create relatively sophisticated interactive multimedia

applications without writing scripts or programming code. Authorware also provides the ability to access routines that reside external to it. These routines are written in languages such as Pascal, C, and HyperTalk (including most HyperCard XCMDs) and can be used to access databases, or control devices such as CD-ROM drives. Authorware has built in videodisc controls, and can import QuickTime movies and animations created in Macromind Director. One of its main strengths is its ability to track user activities and record responses to questions, both of which are extremely important when working with computer-based-training. Authorware, however, lacks its own scripting language which can complicate matters when additional functions need to be programmed into the icons.

Authorware supports cross-platform development. The Windows version is a stand alone application that can be used to author directly on the Windows platform. Applications developed on the Macintosh can be converted for playback in Windows.

HyperCard. HyperCard was one of the first hypertext authoring tools to be widely distributed. HyperCard is based on a card and stack metaphor, a model which has strongly influenced the design of hypermedia applications. Its scripting language, HyperTalk, is still the standard against which others are evaluated. It is a high level object-oriented programming language which can be extended with XCMDs (external commands) and XFCNs (external functions). Although HyperTalk is powerful, it is relatively easy to learn. Overall, HyperCard presents a strong scripting environment for authoring hypermedia applications, however, it is severely limited by its current lack of support for color.

Macromind Director. Director was originally developed as an animation program and continues to be one of the most powerful animation tools available on the desktop. With the addition of Lingo, a HyperTalk-like scripting language, Director has also moved to the forefront of multimedia authoring. Lingo is a high-level object oriented programming language that can be used (in combination with Director's animation tools) to create highly complex interactive applications. Like HyperCard, Lingo cannot directly control external devices, but instead calls external routines, known as Xobjects, for this purpose. Lingo can access external routines written in languages such as Pascal, C, and HyperTalk, including most XCMDs. This ability to reach outside of its native programming language lets authors extend Director's capabilities to include access to relational databases and complex on-screen devices for controlling interactions. Director also includes a reasonably good 32 bit paint program and a very basic set of object-oriented drawing tools (no bezier curves or free-hand line tools).

Director provides relatively sophisticated control over sound, allowing two separate channels to be accessed and controlled independently, but played simultaneously. Sounds can be imported as digitized sound files, or lingo can be used to access and control MIDI (Musical Instrument Digital Interface) sequences. It can also import and play QuickTime movies. One of its greatest weaknesses however, is its text handling ability. This is especially apparent when a lingo script is used to load the contents of the text field. Director provides good support for cross-platform development, however in Director's case, the application must be developed on the Macintosh and then converted using the Director Player for Windows. The Windows Player supports Lingo so interactive movies can be distributed cross-platform.

Digital Sound

Sound is an element previously not part of the cartographer's repertoire, however, cartographers are now beginning to look at how sound can be added to maps to enhance and increase the map's message. Sound is analog information. In digitizing it, sound (or more accurately sound waves) are sampled periodically. Each sample is like a snapshot of the sound at that moment. The frequency at which the samples are taken determines the quality of the sound, in much the same way that the number of frames per second used in recording and playing back motion video affects the quality of an image. The minimum sampling rate necessary to reproduce most musical instruments accurately is 40,000 per second, or 40 kilohertz (40 kHz). For audio Compact Discs waves are sampled at a rate of 44 kHz. Digital Audio Tape (DAT) uses a sampling rate of 48 kHz, a Macintosh samples at 22 kHz, and the lowest recommended rate for quality speech is 7 kHz.

Digital audio can require a tremendous amount of storage space, and the higher the sampling rate, the larger the file. For example, one minute of digital audio sampled at 48 kHz uses 2,813K, sampled at 22 kHz it would require 1,304K, and at 7 kHz it would need 435K. Many multimedia authoring tools can incorporate sound into a presentation or application, however, the shear size of the files can often create sluggish performance. As an alternative to digital sound, some applications (e.g. Director) can access Musical Instrument Digital Interface (MIDI) sequencers. Sequencers are programs that store sets of instructions that are used to produce sound on synthesizers residing external to the computer. Since the sequencer stores only the instructions to make sound, and not the actual digitized sounds themselves, the files are minuscule in comparison.

Graphics and Compression File Formats

Most cartographers have dealt with vector based drawing programs and PostScript or EPS graphics file formats. However, as cartographers begin to work with multimedia they will be confronted with a bewildering number of raster-based formats that are often incompatible. There are two types of programs that can help with this: paint/photo-editing and graphics file translation.

Paint programs work primarily with bit maps and give you pixel by pixel control over the size, resolution and look of the image (although some such as Adobe Photoshop, allow you to work with vector drawing channels that are eventually converted to bit maps). With a program like Photoshop you can import a file in one of several formats; resize it; change its resolution; change its bit depth; crop, retouch or alter sections of it; then save it in one of several different formats and compression schemes for importing into for example, an animation or authoring program.

Graphics file translation programs literally take files in one format and translate them into another. One, Equalibrium Technologies' DeBabelizer, goes a step further. It not only translates virtually every bit map file format available on the Macintosh, DOS/Windows, UNIX, Atari, Amiga, SCI, and Apple II platforms, it also offers scriptable batch processing and 8 bit custom palette creation. Custom palette creation can be especially valuable in multimedia because it allows images at an 8 bit color depth to appear on the monitor at, or close to, 24 bit quality. Since the images are only stored at 8 bits, the quality of the image is preserved on a large number of systems (i.e. does not require a 24

bit monitor card), and the speed of interactive, animated presentations is dramatically increased.

Usually a number of different application programs are used in creating a multimedia presentation, and often each application can create, import and export only a limited number of file formats. Bit mapped or raster based graphics formats are the predominant file format in multimedia. Since bit map graphics store information inefficiently relative to postscript, image compression (making file sizes smaller) becomes a concern. Different file formats support different algorithms for compression. Finding the formats and compression schemes that will be compatible with all the necessary programs can be tricky, especially if the presentation is intended to be delivered cross-platform.

Joint Photographic Experts Group (JPEG). Developed by the Joint Photographic Experts Group, JPEG is a compressed bit map file format that is predicted to soon become the dominant format for storaging scanned photographs. JPEG is a lossy encoding system, in other words the decompressed image will not be identical to the original. However, the reconstructed image often equals or exceeds the representations offered by other exchangeable file formats. Rather than storing specific color information for each pixel, JPEG stores information on color changes. By replicating accurately these changes, the reconstructed image appears to be visually similar to the original. Using this approach, JPEG can achieve extremely high compression ratios (around 24:1) without noticeable degradation of the image. In other words, an image that was originally stored using 24 bits per pixel can be compressed down to about 1 bit per pixel or less. Compression ratios as high as 100:1 are possible depending on the image. This can be very useful when files are transferred over a network, or in multimedia applications distributed on media such as CD-ROM where data transfer rates are still desperately slow. The file is transferred in its compressed form, thereby reducing the transfer rate by roughly the amount of the compression (for example, 24:1). JPEG is supported by PICT, QuickTime, NEXTstep 2.0, variations of GIF, and it is proposed that it be included it in the next release of TIFF (6.0). Its primary disadvantage is that as a software compression and decompression scheme, it is relatively slow. However, the speed is continually being improved and performance can be dramatically increased by adding NuBus boards dedicated to performing the JPEG algorithms.

LZW. Built on work done by Lempel and Ziv, T.A. Welch developed an algorithm that works on the principle of substituting more efficient codes for data. A binary code is assigned to each unique value in the file, the shortest codes are assigned to the most frequently occurring values. These code assignments are stored in a conversion table that is accessed by the decoding software. LZW achieves modest compression ratios relative to JPEG, ranging from 1:1 to 3:1. However, since each value in the original file is represented in the table, the original file can be reconstructed exactly. Consequently, LZW is considered a lossless compression scheme. It is supported primarily by GIF and TIFF file formats.

Motion Picture Experts Group (MPEG). MPEG is a standard related to JPEG that is used to compress and store digital video. In effect, an MPEG file contains series of JPEG-like frames along with information that assists decompression software, such as Apple's QuickTime, to interpolate the individual frames.

Graphics Interchange Format (GIF). Developed by CompuServe, GIF is a file format that is used for the exchange of bit map graphics files between different systems on the CompuServe network. It is generally not the primary format for any computer graphics programs. It is limited to 8 bit graphics, but can create custom 8 bit palettes for 24 bit images. GIF can not store CMYK or HSI color model data, color correction or grayscale data. Nevertheless, it is a very useful format for exchanging files across different platforms as it is supported by most desktop computers and UNIX-based workstations.

PCX. Originally developed as a proprietary format for ZSOFT's PC Paintbrush program, PCX is now widely supported by DOS and Windows applications, including most scanners. PCX can support files containing up to 24 bits of information per pixel.

Microsoft Windows Formats. As part of the strategy to develop multimedia on the Windows platform, a relatively new bit map format has emerged: the Microsoft Windows Device Independent Bitmap (BMP/DIB). This format, similar in structure to PCX, is intended to introduce to Windows applications the kind of interconnectivity provided by the PICT file format on the Macintosh. It is capable of supporting bit maps with up to 24 bits per pixel.

PhotoCD. PhotoCD is the proprietary bit mapped image format and compression system used in the Kodak Photo CD system which incorporates a dedicated 35mm slide scanner, computer system and multisession CD-ROM pressing drive. Slides are scanned, color corrected, compressed using Kodak's proprietary YCC compression, and recorded to the CD-ROM in one operation. Each single session Photo CD can hold approximately 100 slides. However, because Photo CD pressing drives are multisession, fewer slides can be scanned in the initial session and additional slides can be added later to the same CD. Multisession Photo CDs hold slightly fewer slides depending on the number of sessions. Kodak is planning to release five different formats of Photo CD: Photo CD Master, Pro Photo CD, Photo CD Portfolio, Photo CD Catalog, and Photo CD Medical. At present, only Photo CD Master is available. In this format, each slide is scanned and stored in 5 different resolutions. Photo CD is supported on both the Macintosh and Windows platforms, and when used with a special Photo CD player, the images can be viewed (with limited pan and zoom features) on a standard video (television) monitor.

PICS. PICS is a file format used primarily on the Macintosh platform for storing and exchanging animations. It is a sequence of PICT images that begins with the first frame or image area of the animation and continues through the changes to that area. Although a PICS file contains all the individual frames of an animation, the frames are not stored as separate files. Many programs, such as Macromind Director, Authorware Professional, Aldus SuperCard, and Swivel 3D can export and import animations in PICS format. Consequently, it is becoming the defacto standard for transferring animation files on the Macintosh.

QuickDraw Picture Format (PICT). PICT was originally developed by Apple Computer, Inc. as a standard for interchanging QuickDraw pictures between Macintosh programs. However, many DOS based multimedia applications have begun to support PICT. As such, PICT is emerging as a viable cross-platform file format. PICT supports vector and bit mapped graphics, as well as binary page description. In addition, it supports 32 bit color graphics files and JPEG compression. In turn, PICT and JPEG are both supported by QuickTime and QuickTime for Windows. This combination makes PICT a versatile format that is especially useful for distributing multimedia applications on both the

Macintosh and DOS platforms. Both Macromind Director and Authorware Professional use PICT as their primary graphics file format, and it is the only format that they can import.

QuickTime. Technically, QuickTime (developed by Apple Computer, Inc.) is not a file format, it is a system software extension for both the Macintosh and Windows platforms. Nevertheless, it presents a standard "format" for creating, storing, and interchanging time-based information such as digital video, animations and sound. In effect, it is intended to do for time-based information what QuickDraw does for graphics; establish a display standard that can be accessed by all application programs. QuickTime actually consists of four major components: system software, standard data formats, compression software, and interface guidelines. The availability of a standard such as QuickTime allows the same time-based information to be shared across platforms by numerous application programs running on a wide variety of computers, and at the same time provide the user with a consistent interface for accessing that information.

Tagged Information File Format (TIFF). TIFF was developed jointly by Aldus Corporation and Microsoft Corporation as a file format for bit mapped graphics. It can accommodate black and white, grayscale and 32 bit color, and is generally considered one of the best formats for storing scanned photographs. It supports LZW compression and is widely accepted on the Macintosh, DOS and UNIX platforms.

Truevision Targa. Targa is a format developed by Truevision, Inc. It was one of the earlier formats for storing 24 bit image files and consequently, is widely supported on the DOS and Macintosh platforms.

Scanning

Scanning is a common way to convert analog images into digital information. In cartography this is most often done when information from a paper basemap is need in digital form. The scanned image is affected by how a scanner senses light, color, and resolution.

Charge-Coupled Device (CCD). The chips that sense light reflected or transmitted from the image being scanned are called charge-coupled devices (CCDs) and are usually combined into linear arrays. The more CCDs in the array, the higher the linear resolution of the scanner. In most scanners, the array is moved over the image line by line (the width of one sensor). However, with scanners such as sheetfed scanners, the image is passed in front of the array. Each CCD element captures information for one picture element (pixel), one line at time, in color bit depth resolutions ranging from 1 to 48.

Color Bit Depth. The amount of information (the number of bits) that is sampled by the CCD element for each pixel is referred to as the bit depth. For example, black and white scanners sample each pixel of the image and based on a predetermined threshold, assign a value of either 1 (white) or 0 (black). Black and white scans are said to have a bit depth of 1. Most scanners are capable of scanning at least 8 bits per pixel. In binary this yields a range of 0-255 which can be used to record one of up to 256 possible values for each pixel. This is more than sufficient for representing gray scale information. Almost all color scanners are based on an RGB (Red, Green, and Blue) color model and sample data for each of the Red, Green, and Blue channels. In order to represent continuous tones for RGB then, the scanner must sample at 8 bits per channel, or a bit depth of 24. This produces up to 16.7 million binary combinations and

consequently, each pixel can be assigned one of 16.7 million different colors. Although computer systems generally cannot use image files containing more than 24 bits per pixel,[1] scanners that collect more information per pixel (10 to 16 bits per channel or 30 to 48 bits per pixel) generally interpret the data and output a file that contains only the most useful 24 bits. This additional information is especially helpful for bringing out the details in shadows as in the case when lettering crosses line work on old maps.

Scanning resolutions. The true resolution of a scanner is determined by its CCD array. For example, a typical 8.5 inch long array uses 2550 CCD elements to achieve a true resolution of 300 pixels per inch (2550 ÷ 8.5 = 300 ppi). To increase this resolution, more elements must be added to the array, thereby increasing the cost of the scanner. Many lower cost scanners boost the "resolution" without adding additional elements by means of software interpolation. The software, in effect, creates additional pixels by taking each sampled pixel and dividing it into two or more smaller ones. For example, a 300 ppi scan can be boosted to 600 ppi by taking each sampled pixel and dividing it in two. The values given to these created pixels is determined by interpolating the values for the surrounding sampled pixels. Although interpolation can often increase the sharpness of a scan by reducing the jaggies, it does not achieve the level of detail captured by an equivalent number of CCD elements. Consequently, for images that are primarily line art (often the case with maps) interpolation can significantly improve the resolution of the scan, however, for continuous tone images, interpolation may improve the scan only slightly.

One of the most important questions concerning resolution is "How much is enough?" Traditionally, scanning resolutions have been determined by using a formula driven by the desired screen resolution, given in lines per inch (lpi), that will be used in printing. These formulas, however, are of little use for determining resolutions for scans to be incorporated into multimedia productions. In multimedia, the output device will generally be a computer or television. Most computer monitors (e.g. the Macintosh 13" RGB, and SVGA monitors) have a screen resolution of 640 pixels (approximately 9" horizontal) by 480 pixels (approximately 6.5" vertical) at a color depth of 8 bits per pixel. This works out to approximately 72 ppi. For a 100% enlargement then, the scan need only be made at 72 ppi.

Scanners

A variety of types and forms of scanners are available. The decision to use one over another is based on cost, size of the image being scanned, medium of the original, color, and resolution.

Hand-Held. Generally, the least expensive are small hand-held devices typically with a horizontal scanning width of about four inches. Hand-held scanners are of limited value because the scans are made by dragging the device (the linear CCD array) over an image, often resulting in uneven, poorly registered scans. Also, the limited width of the linear CCD array requires two or more passes for images wider than about four inches. The two or more resulting digital files then need to be "sewn" together or edge matched, a cumbersome process that can be especially frustrating with line work such as maps.

Flatbed. Although flatbed scanners range in linear resolution from 300 to

---

[1] Actually computer systems can work with up to 32 bits of information, but 8 bits are reserved for an alpha channel. The remaining 24 bits are used for the color information.

over 1200 ppi at bit depths from 8 to 30, the ones under $10,000 generally range from 300 to 800 ppi at 8 to 24 bits. In most flatbed scanners, the artwork is placed face down on glass and the CCD array and light source are passed underneath, much like the typical photocopy machine. The quality of the scanner is determined by the precision with which the array and light source are moved, the resolution of the CCD array, and the software that interprets the information. Almost all flatbed scanners capture color information using the RGB color model. Newer models capture all three channels in one pass by using three light sources (one red, one green, one blue) during the scan. However, many of the older models use only one light source and pass the array over the image three times, once each with a red, green or blue filter in place. Most flatbeds can scan originals up to 8.5 by 11 inches, some high-end models can scan up to 11.7 by 17 inches. In general, flatbed scanners provide a relatively high quality scan at a reasonable price. Flatbed and Sheetfed scanners are usually used for Optical Character Recognition (OCR) scanning.

Sheetfed. Sheetfed scanners are similar to fax machines where rollers draw the image into the scanner and past the CCD array and light source.

Overhead. Overhead scanners suspend the CCD array above the image or object to be scanned. This allows objects to be captured in 3D, however, because the light source is projected onto the object, the quality of the scan is diminished.

Slide/Transparency. Slide/Transparency scanners project light through the film to be scanned and into the CCD array as it is moved across the film. The projected light provides more saturated colors and greater detail than is achieved with reflected light. Some flatbed scanners can be fitted with a transparency option, however, they tend to produce lower quality results and generally cost as much or more than dedicated slide scanners. Dedicated scanners provide resolutions ranging from 1000 to 6000 ppi at bit depths of 24 to 48.

Drum. Drum scanners represent the high end. Not only do they produce the highest quality scans, they are also the most expensive (generally over $70,000 but recently as low as $40,000). With these scanners, the image or film is mounted on a scanning drum and a focused beam of light is moved across the image. The light can be used to expose photographic film mounted on an exposing drum, or to create a digital file. The level of precision at which the focused light source is moved yields extremely high quality scans.

Storage Media

Storage media can be distinguished based on its capability, receptivity (magnetic or optical), and capacity. Magnetic drives use disks which store information coated with a material that is receptive to magnetic encoding. Optical disc[2] drives use discs which store information of extremely high density in the form of pits that are written and read by a laser beam. Some optical drive mechanisms are used to record analog data to the discs, others record digitally encoded data. A variety of both magnetic and optical storage devices are available.

CD-ROM Drives. CD-ROM (Compact Disc-Read Only Memory) drives utilize the same basic technology originally developed for audio compact discs.

---

[2] Typically a spelling distinction is made between the storage devices of magnetic and optical media. Magnetic disks are spelled with a "k," whereas optical discs are spelled with a "c."

In fact, CD-ROM drives can play standard audio CDs, or CD quality audio tracks encoded onto CD-ROM data discs. A CD-ROM disc is only slightly larger than a high-density floppy disk, but can hold over 450 times more information depending on the format used in mastering the CD. Most CD-ROM drives can read discs mastered for slightly over 650 megabytes, others can only read discs that have been mastered for about 580 MB. CD-ROM discs are Read Only Memory and cannot be written to or created without specialized equipment which, until recently, was cost prohibitive except for large dedicated pressing facilities. However, recent advances in CD-ROM pressing technologies have brought down the cost of creating single copies of CD-ROMs (one-offs) to where they provide an extremely competitive solution for back-up and distribution of large quantities of data to a single location.

Hard Disk Drives. The drive mechanism of a hard disk drive consists of one or more non-flexible fixed disks, two magnetic read/write heads per disk (one for each side), a housing, and an electronic interface for connecting to the computer (ST-506, SCSI, ESDI, or IDE). Drive capacities range from 10 MB to 3,000 MB with access times ranging from 10 to 50 milliseconds. The read/write heads float on a cushion of air over the disks that spin at 3600 rpm. Electronic signals received from the computer (via the interface) activate motors (either stepper or voice coil) that position the heads for either reading from or writing to a specific address (location) on the disk.

Removable Cartridge Drives. These drives are essentially 5 1/4" magnetic, hard disk drive mechanisms that allow the disk (housed in a plastic or metal cartridge) to be inserted and removed like a floppy disk. Disk capacities range from 20 MB to 150 MB, although the most common are 44 MB and 88 MB. Drives tend to be only slightly slower than fixed platter drives. Generally, cartridges can be interchanged among drive mechanisms of the same format and manufacturer, thus making the transportation of large files (generally the case in multimedia applications) between computers with compatible cartridge drives quite easy. Some drive mechanisms such as the Syquest 88c or the Bernoulli MultiDisk 150 can read and write to more than one capacity of cartridge.

Removable Optical Drive. Removable Optical Drives have storage capacities of up to one gigabyte that allow the same area of an optical disc to be recorded to and erased multiple times. This brings to optical drives the convenience and flexibility of magnetic hard drives, however, optical drives tend to be much slower. Recently, removable 3.5 inch cartridge drives have gained popularity. Depending on the drive mechanism, they can read and write to cartridges of either 128 MB or 256 MB. These cartridges can then be interchanged with other removable drives, making them convenient for transporting large files to and from locations such as service bureaus.

Write Once Read Many (WORM). Unlike CD-ROMs which require pre-mastering and mastering before usable copies can be made, WORM drives allow one time recording of data directly from the computer. Once the data is written to the disc (which can occur over multiple sessions), the drive becomes read-only storage media. Worm drives can have storage capacities up to 1 terabyte. However, with the recent advance of a number Compact Disc Recordable (CD-R) drives for under $5000, the popularity of traditional WORM drives has diminished. CD-R is in reality a cross between CD-ROM and WORM technology. The advantage of CD-R is that it records to standard CD-ROM discs which can be used on most CD-ROM drives, thereby greatly increasing the distribution potential.

Video

Video has traditionally meant the picture phase for television broadcasting and has been distinguished from audio which deals with sound. Multimedia makes use of both analog video (traditional video) and digital video.

Analog Video. Video, in the form of television and movies, is analog. The analog signal is comprised of a wave form varying in frequency. Videodiscs (12" reflective optical discs) and video tape store images in analog format.

Digital Video. Whereas analog signals vary in voltage, digital video converts analog signals to discrete bits of information. In its broadest sense, it covers a number of activities ranging from capturing still images to digitizing and editing full-motion video. Still image capture is a form of scanning that utilizes a video camera instead of a CCD array. The camera is pointed at a stationary object or image and the signal is passed through the camera to a computer program that coverts the analog video signal to a single digital graphics file. A variation of this is called a frame grabber. Here individual frames from a motion video source are grabbed and digitized, thereby allowing motion video to be digitized. However until recently, most frame grabbers stored the captured frames in RAM, usually a severe limitation on the number of frames that could be digitized. Now, efficient image compression algorithms (such as the standard developed by the Joint Photographic Experts Group (JPEG)) and protocols for synchronizing time-based digital information (such as Apple's QuickTime) have begun to make the real-time digitizing and playback of motion video a reality on the desktop. A number of new frame grabbers, such as SuperMac's VideoSpigot, utilize both JPEG compression and QuickTime protocol standards to achieve acceptable recording and playback of motion video. Once the video has been digitized, editing software such as Adobe's Premiere, provide the ability to integrate the video with a additional media such as animations, photographs, graphics and sound. The resulting integrated linear segments can be played back as stand alone presentations or incorporated into documents created by applications such as word processors, charting and presentation programs, or multimedia authoring tools.

APPLYING MULTIMEDIA TO CARTOGRAPHY: THE INTERACTIVE VIDEODISC MAPPING PROJECT

The remaining portion of this paper describes the Interactive Videodisc Mapping Project carried out at the University of Wisconsin-Milwaukee. The project was funded by a grant from the United States Department of Education, College Library Technology and Cooperation Grants Program. The project had two basic objectives. The first was to use multimedia technology to create an interactive videodisc to illustrate the broad and diverse topic of mapping. Second, but no less important, was to provide access to some of the rare and valuable examples of cartography from one of the premier map libraries in the country, the American Geographical Society Collection.

Description of the Videodisc

The double-sided videodisc contains on one side, 28,900 still frame images of more than 600 maps and a four minute motion video documentary on the procedures used to create the videodisc. The second side has three short (approximately 7 minutes each) full motion narrated video segments: "Changes

Through Time" illustrates how maps can communicate information about the past as well as show trends about the future; "The Cartographer's Choice" explains how some mapping decisions are made; and "Maps, You Gotta Love Em" is a fast paced look at different types of maps and their various uses. A fourth video segment (also approximately 7 minutes) demonstrates how the videodisc images and database might be used.

All of the images on the videodisc are linked to a digital database (designed for the Macintosh) containing information about each map's title, region, scale, projection, date of publication, etc. For each map in the database there is one image of the entire map on the videodisc as well as a series of systematic enlargements (tiles) and close-up images (enlargements of selected areas of interest on the map). The images for each map are linked to information in the database by a unique object identification number. The database also links the videodisc images to other related images on the videodisc as well as to other information in the database.

The videodisc and database are designed to work together in an interactive, multimedia environment. The videodisc images are analog and played on a television (video) monitor from a videodisc player. The database information is digital and read from a floppy disk by a Macintosh computer and displayed on a computer monitor. The ideal arrangement when using the videodisc and database is to have the television and computer monitor side-by side (or projected side-by-side) to enable viewing an image and its information at the same time. The user interface, specifically developed for this project, is what allows users to link the database information for each map to the appropriate images and permits database searches. The link is bi-directional so that images can be accessed via the database, or the database information can be accessed via the images.

Analog vs. Digital Technologies

One of the first issues in developing this multimedia tool was deciding on the appropriate technology and media for displaying images of maps. Amongst the many concerns was attaining a high quality image that would capture the rich color and detail present on maps. One option was to reproduce the maps on a CD-ROM. The map images and their data would be stored in a digital format and displayed on a computer monitor. Although this medium would have offered a number of advantages, CD-ROM technology was still in its infancy at the time the grant proposal was developed (1990). The large size of the original maps,[3] restrictions on the number of images that could be placed on a CD-ROM, and concerns over image pixalization and color reproduction quality were also problematic. An added concern was how the data and map images would share screen space if they were both being played off the same media.

A second option (and the one decided on) was imaging the maps onto a videodisc and creating an accompanying digital database. Videodisc (or laserdisc) technology relies on an analog signal and displays images on a standard video monitor in an NTSC (National Television Standards Committee) display. We were encouraged by the number of successful projects that had

[3] Since most of the maps intended for the project were larger than page size (in some cases as large as 4' x 6'), scans on a typical flatbed scanner would have been prohibitive and an intermediate slide or color transparency would be required.

used videodisc technology and reports and studies showed that this media was an effective way to capture and display thousands of images. The image storage capacity of a videodisc is truly exceptional. Each videodisc side is capable of storing 54,000 individual images or 30 minutes motion video (or a combination of the two). Most advantageous, however, was the high level of interactivity that could be achieved with a videodisc and the almost instantaneous access to images regardless of their position on the disc.

Selection of Map Images

The maps selected for inclusion on the videodisc were based on the following criteria:

1. A map's representativeness of a cartographic characteristic or map type. For example, a 1972 plat map by Rockford Map Publishers was used as a typical example of a map that show detailed land ownership. A 17th century Dutch atlas was selected because it was a typical example of coloring techniques used for maps during that period.

2. A map's exemplary portrayal of a cartographic characteristic of map type. For example, National Geographic Society's 1992 map of Amazonia was selected for its unique and innovate design and cartographic style that incorporates text, graphics, and maps in a single sheet.

3. A map's temporal, regional, or topic relatedness to a previously selected image. For example, we selected a 1789 plan of Naples and also a 1990 plan of the same area to illustrate how mapping styles (and the landscape that was mapped) had changed. In addition, two maps of Philadelphia, corresponding to the dates of the Naples maps, were selected to show how mapping practices and styles differ regionally.

4. A map's uniqueness as a rare map. For example, we selected Leardo's *Map a Mundi* made in 1452 because there are only three known original copies, allowing the videodisc viewer access to a map they might otherwise never be able to see in such detail.

The 600 plus maps cover every region of the world, with scales ranging from a map of the interior of a shopping mall to a map of the universe. There are over seventy different map projections represented on the videodisc. There are more than ninety map subjects ranging from maps of administrative divisions to whaling charts. There are maps published by all levels of government as well as private publications from all over the world. There are maps from atlases, newspapers, magazines, map sheets, and books. There are maps that are hand colored as well as some made by computers or satellites. There are maps of the earth during the pre-Cambrian period, maps of the exploration of the Americas, and an animated map that shows what the population of the earth might be in the future. There are maps etched on paper, printed on fabric, and even made out of cheese. In all, it is possible for a user of the videodisc to search a variety of characteristics and find a rich assortment of images to look at and study.

Map Imaging Process

The NTSC display of the standard video monitor is not capable of storing an

image of a large map at a resolution which preserves critical map details. To compensate for the low resolution, analog map images were recorded directly onto a re-writeable optical disc cartridge via a videodisc recorder and video camera. The high-resolution recording mode stores images at a video resolution in excess of 400 lines and much of the map detail was preserved by using a high quality video camera. A technique was developed whereby an image of every map was taken in its entirety and systematic enlargements (tiles) were taken at a resolution appropriate for the specific detail of the map. The tile sizes on average covered a 3 cm x 4 cm area of the map. A given map could have as many as 200 tiles or none depending on the original map dimensions and detail (the average was approximately 40 tiles per map).

The video feed to the videodisc recorder was through a Sony DXC-930 3-CCD video camera with a 12 to 1 zoom lens. The camera and lens configuration provide sharp images through three .5" IT Hyper HAD CCD's, each with 380,000 effective picture elements with a resolution of 720 TV lines. The camera signal also provided a feed to a video monitor in order for the camera operator to preview what was to be recorded on the optical disc cartridge.

The imaging process involved moving the maps under a vertically mounted camera rather than moving the camera. The maps were laid flat on an imaging platform that was attached to a motor driven wheeled apparatus which allowed four feet of movement in the "Y" direction and six feet of movement in the "X" direction. The map was centered under the camera and a picture of the whole map was recorded onto the re-writeable optical disc cartridge. The tiles and close-up images required precise and accurate positioning under the camera and zoom control of the lens. Computer controlled stepper motors dictated the movement of the imaging platform in both the X and Y direction (to within one millimeter) to position the map under the camera for the tiles and pieces. When each tile position was reached a signal was automatically sent back to the videodisc recorder to record the image before the imaging platform moved to the next tile.

The User Interface

As with any multimedia application, the user interface is a critical aspect of the planning and development. The primary function of the interface for the videodisc was to link images on the videodisc with the information about the map it belongs to in the database. This is a bi-directional link that allows images to be accessed via the database, and the database to be accessed via the images. The initial proposal envisioned development of a cross-platform interface for use on both a Macintosh and DOS platform. Despite the claims of dual platform authoring software, we found the task impractical given the time and budget constraints.[5]

The interface was structured around four display environments:

- **Map Information Display:** This environment allows users to view information for individual map records and access videodisc images that correspond to that information.

[5] The primary problem was the lack of existing external commands (XCMDs) to implement specialized videodisc controls on a dual platform. This was further complicated by the need to integrate authoring software with the relational database program.

• **Image Browsing:** This environment permits users to browse images on the videodisc and place the images on a Request List that they can name and save to their own floppy disk.

• **Database Search:** This environment provides the ability to search for maps with specific characteristics, obtain lists of all the maps on the videodisc that have that attribute; view one, any, or all of those maps; and save their titles to a Request List.

• **Supplemental Information Display:** This environment contains a dictionary of terms, chapters on specific map subjects, and digital illustrations to enhance explanations.

The user is presented with a choice of three activities when they start the interface videodisc program. They can ***Browse Images*** on the videodisc, ***Search the Database*** for specific characteristics to yield a list of all the maps on the videodisc with those characteristics, or ***Browse a Collection*** containing prescribed subsets of maps with a coherent theme that we have created.

If they select ***Search the Database*** they are presented with a finder similar to the Macintosh System 7 finder. The finder is designed to allow the user to query the database by asking it questions. By clicking on the left hand entry field of the finder, all the categories of map characteristics in the database are listed. If the user was looking for *Spanish Language* maps made before *1600* they could choose ***Language*** from the available map characteristics. This would configure the appropriate operator (***contains***), then they would choose ***Spanish*** from the pull down menu of choices to the right. They could continue with the second parameter in the same fashion by selecting ***date*** on the second line, using the operator ***before***, and typing in the date ***1600***. In all, four parameters can be searched at one time with designators of 'and' or 'or' and with characteristic operators such as ***contains, is, is not, does not contain, before, after,*** etc.

When the designated parameters have been specified the user would click on the ***Find*** button and the screen would change to one displaying all titles of all the maps that fit their search criteria in a window on the left and the ability to create a Request List as a subset of the maps if they chose to do so. The user could click on any of the map titles and view the videodisc picture of that map on the television monitor. They could also double click on one, many, or all the map titles to place them on a Request List. The request list can be named, saved on a personal floppy disk, and used at anytime to recall and access the maps on it. Double clicking on any title once it is in a Request List will bring up the Map Information Display environment containing all of the information for that specific map.

This display environment allows users to view information for individual map records and access videodisc images that correspond to that information (see Figure 1). The primary components of this screen are the Tile Access Grid (upper left), the Close-ups Listing (middle left), Buttons for accessing the primary map record information (bottom), and a screen area for displaying that information (text area on right). Also included are the abbreviated map title (upper right), the map's database Object Number, and the Frame Number for the current videodisc image (upper left). A Menu Bar along the top of the screen provides access to ***File*** and ***Edit*** functions; ***Search*** (access to the ***Image Browser*** and ***Database Search*** environments); a ***Dictionary; Request Lists; Links*** (links

between this and other map images and related information); and an ***Activity Log*** (that monitors images accessed).

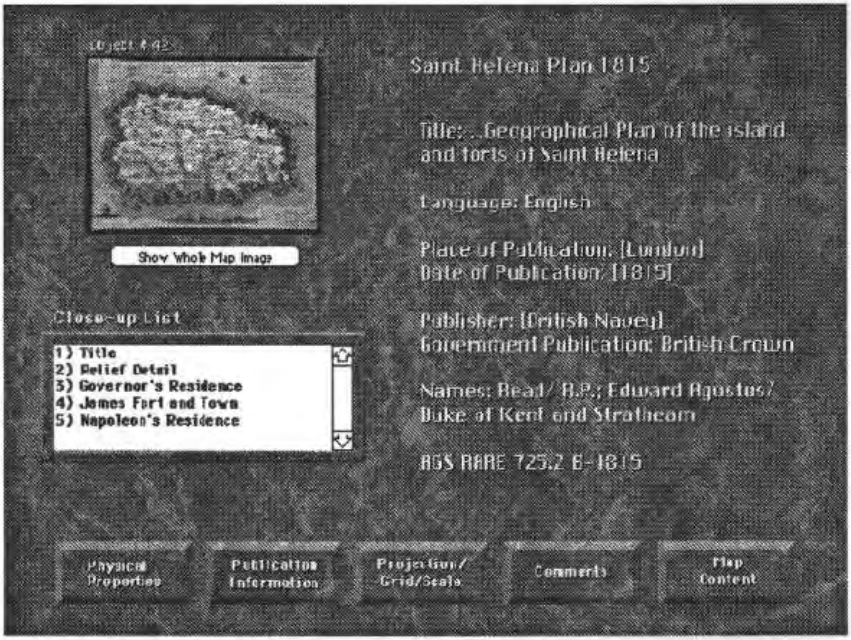


Figure 1. The Map Information Display environment showing the *Publication Information* for St. Helena Plan of 1815.

The database information for the primary map is divided into five general categories and accessed by the buttons. Users can access the information for each of these categories by clicking the appropriately labeled button. ***Physical Properties*** includes the color scheme, execution, medium, and original map dimensions. ***Publication Information*** includes the map title, edition, language, place of publication, publisher, date of publication, whether it is a government or other type of publication, the authorities involved in the map, if it is from an atlas or book, its AGSC call number, and its OCLC number if it has one. ***Projection, Grid & Scale*** information includes the map scale, grid designation, its Prime Meridian, and projection. Each map record contains ***Comments*** that provide a brief written narrative about the map when appropriate, or refers the user to other images on the videodisc or other sources of information. ***Map Content*** includes the region covered on the map, its latitude and longitude extent, date of situation, subject, mode of representation, and cartographic themes that we have chosen to highlight.

The Tile Access Grid has a coarse digital background image of the map overlain by a grid. The grid indicates the number and location of the tiles available for that map. The user can access videodisc tiles by placing the mouse pointer over the Tile Access Grid. Pressing the mouse button and positioning the pointer over a tile in the grid brings up the corresponding image from the

videodisc onto the television monitor. Pressing the mouse button and moving the pointer over the grid simulates panning the map and the speed at which the videodisc displays the corresponding image is directly related to the speed the mouse is moved. What makes the Tile Access Grid work is that the number of tiles for each map, the videodisc frame number for each tile, and the centroid coordinate for each tile are in the database. This data is generated before the map is ever imaged. It is the same data that controls the movement of the imaging platform.

In addition to its tile access function, the Tile Access Grid shows the size and position of each close-up with a rectangular symbol labeled with numbers that correspond to the Close-up Listing. The Close-ups Listing is a scrolling window that lists the number and short description for each close-up. The user can click on the description and the information for that close-up is displayed in the information display area and the close-up image is displayed on the television monitor.

Users also have access to an *Activity Log* that records the date, time, and title for every map or image accessed in a single session. If a user wanted to look at a previously viewed map they can access the activity logger, click on the map title of the map they wish to see and it is automatically displayed on the television monitor.

CONCLUDING REMARKS

We have attempted to provide readers of this paper with an example of a multimedia, cartography application and a glossary of some multimedia terms. The following concluding remarks, though brief, warrant attention by cartographers.

Much of what is driving the development of multimedia tools is the idea that graphics are an extremely effective and powerful way to communicate information. This is not something new to cartography– in fact, it is mainly for this reason that multimedia lends itself so well to communicating cartographic information. Multimedia offers cartographers exciting opportunities to work with graphics in new and different ways. As we begin to experiment and understand these new tools we should keep in mind that not only are they beneficial for improving the methods and processes we already employ, but that they have great potential to change the way we conceive of maps.

We should look to multimedia for mapping solutions that not only enhance the way map information is communicated, but for the ways in which it can change the look of maps. This requires moving beyond the use of multimedia components individually. Instead, these components should be harmoniously integrated to communicate information in new and different ways. For example, instead of simply using animation to show spatio-temporal change (animating a series of static maps), animation can be combined with sound and interaction to change the basic nature of how relationships are communicated. It will be through this type of integration that maps and the communication of spatial information will reap the greatest benefits.

SELECTED BIBLIOGRAPHY

A variety of reference materials that address multimedia applications and products are available. Generally, information about the latest developments

are found in the professional trade magazines. Listed below are a limited number of some of these sources as well as a few reference books on the topic.

Apple Computer, Inc. 1992, *Apple CD-ROM Handbook*, Addison Wesley, Reading.

Busch, D.D. 1992, *The Complete Scanner Handbook for Desktop Publishing: Macintosh Edition*, Business One Irwin, Homewood, IL.

Busch, D.D. 1992, *The Complete Scanner Toolkit: Macintosh Edition*, Business One Irwin, Homewood, IL.

BYTE. Peterborough, NH: McGraw-Hill - vol. 1, no. 1 > Sept. 1975.

CD-ROM Professional. Weston, CT: Pemberton Press - vol. 3, no. 3 > 1990.

Desktop Communications. New York, NY: International Desktop Communications - vol. 1 > 1989.

Kay, D.C. and J.R. Levine, *Graphics File Formats*, Wincrest/McGraw-Hill, Peterborough, NH.

Mac Publishing and Presentations. New York, NY: International Desktop Communications - vol. 1, no. 1 > May - June 1992.

MacUser. Foster City, CA: Ziff-Davis Publishing Co. - vol. 1 > 1985.

MacWorld. San Francisco, CA: MacWorld Communications - vol. 1 >1984.

NewMedia. Riverton, NJ: Hypermedia Communications - vol. 1, no. 5 > July - Aug. 1991.

PC Magazine. San Francisco, CA: Software Communications - vol. 1, no. 1 > Feb.-Mar. 1982.

PC Publishing and Presentations. New York, NY: International Desktop Communications - vol. 1 > 1987.

Yavelov, C. 1992, *MacWorld Music & Sound Bible*, IDG Books, San Mateo, CA.

PRODUCTS SPECIFICALLY DISCUSSED IN THE PAPER

Adobe Photoshop, Adobe Systems, Mountain View, CA (800-833-6687)
Authorware Professional, Macromedia, Inc., San Francisco, CA (415-442-0200)
Macromind Director, Macromedia, Inc., San Francisco, CA (415-442-0200)
Debabelizer, Equilibrium Technologies, Sausalito, CA (415-332-4343)
HyperCard, Claris Corp., Santa Clara, CA (408-727-8227)
QuickTime, Apple Computer Inc., Cupertino, CA (408-996-1010)

AUGMENTING GEOGRAPHIC INFORMATION WITH COLLABORATIVE MULTIMEDIA TECHNOLOGIES

Michael J. Shiffer
Massachusetts Institute of Technology
Department of Urban Studies and Planning
77 Massachusetts Avenue, Room 9-514
Cambridge, MA 02139

ABSTRACT

Many urban and regional planning situations involve informal queries about physical environments such as "What's there?" or "What would it be like if... ?". These queries may be supplemented with a variety of geographically-based information such as maps, images, narrative descriptions, and the output of analytic tools. Unfortunately, many analytic tools, (such as population forecasting models, etc.), lack the descriptive abilities of images and human gestures. For example, while a spreadsheet macro may supply the population density of a given area, it is not able to provide an example of "how crowded the streets will be". Similarly, quantitative representations of environmental impacts, such as noise, may be meaningless to the lay person. This paper explores the implementation of a collaborative multimedia system designed to improve the communication of planning-related information. This is being accomplished through the integration of maps, analytic tools, multimedia images, and other relevant data. The resulting information base is projected on the wall of a meeting room. Participants interact with the system using cordless pointing devices. An implementation of a collaborative multimedia system in Washington D.C. is described. Following that description, the issues surrounding a more widespread implementation of this and similar technologies will be identified.

INTRODUCTION

Planning commission meetings in large metropolitan areas are frequently concerned with the environmental impacts of proposed developments upon the built environment. These situations often require a set of technical analyses that include changing assumptions and priorities, descriptions of significant visual and audible impacts, and may also involve several geographically separated parties such as consultants, developers, and city planners. Furthermore, these meetings often utilize physical, electronic, and cognitive information. Physical information is typically delivered using documents, maps, and images. Electronic information is typically delivered using computer-based information systems (Klosterman, 1992). Finally, cognitive information is delivered using human recollection and story-telling (Forester, 1982). Difficulties are often encountered when accessing these information types. These include organizing physical information in a cohesive and accessible manner, communicating technical information to

non-technical audiences, and the individual orientation of computers. That is, most computers are designed for interaction with one person at a time.

A strategy has been proposed to address the difficulties encountered when accessing information in collaborative planning settings such as the commission meetings described above (Shiffer, 1992). The strategy makes use of a collaborative planning system (CPS) that (1) organizes multimedia information such as renderings, architectural drawings, video clips, and the output of analytic tools in an associative form; (2) provides facilities for the "real time" annotation of maps with text, graphics, audio, and/or video; and (3) aids the representation of normally quantitative information using descriptive images such as digital motion video and sound.

After a brief background description of CPS, this paper will describe a limited implementation of a CPS in Washington D.C. Following that description, the issues surrounding a more widespread implementation of CPS and similar technologies will be identified.

BACKGROUND

Analytic tools such as Geographic Information Systems (GIS) and spreadsheet models have the potential to be tremendously useful to planners for providing forecasts and modeling various phenomena. Yet they can be practically useless to individuals who may not understand how to implement such tools properly. The difficulties encountered in mastering these tools often cause less technically-oriented people to be excluded from the planning process. Thus analytic tools and their outputs need to be made more visually (or audibly) appealing, so that information that would normally be meaningless and intimidating can be made understandable.

A recent technological trend that addresses the need for usable tools has been the development of representation aids for human-computer interaction (Zachary, 1986). Representation aids overcome the need to memorize computer commands by translating the user's actions into commands that can be understood by the machine. This is accomplished by providing a human-computer interface displayed in a form that matches the way humans think about a problem. By doing this, the user can make "rapid incremental reversible operations whose impact on the object of interest is immediately visible" (Hutchins, Hollan, & Norman, 1986, p. 91). Representation aids will not completely replace quantitative measures of environmental phenomena. Rather they will serve to supplement such measures through multiple representations (Rasmussen, 1986). Multiple representations of a problem enable the user to view information in several different contexts thus offering the potential to generate alternative approaches to a problem.

Another common problem that faces users of analytic tools is the individual orientation of their delivery mechanism. While these tools can be brought into collaborative environments with the aid of laptop microcomputers, they are often usable by only one or two individuals to the exclusion of other meeting participants (Stefik, et.al., 1988). Computer-supported collaborative work (CSCW) addresses the drawbacks associated with individual computer usage by putting computer-oriented tools into meeting environments which may be physical, as in small group meetings; distributed, as in local area

networks; or virtual, as in "meeting rooms" found on electronic bulletin board services (BBS). This paper is concerned primarily with small physical meetings of four to ten people. Stefik et al, (1988) espoused the benefits of collaborative computer systems when they noted that computers are often left behind in favor of more passive media like chalkboards and flip charts when meetings are held. Their development of the Colab at the Xerox Palo Alto Research Center (PARC), represents an attempt to demonstrate the collaborative capabilities of computers using a prototype meeting room.

Collaborative Planning Systems (CPS) are designed to take advantage of both representation aids and CSCW in order to present planning-related information in a form understandable to the heterogeneous groups of people that frequently participate in environmental reviews. In addition to this, CPS are designed with an associative hypermedia data structure (Conklin, 1987; Wiggins and Shiffer, 1990) in order to provide a framework for access to non-standard data types such as anecdotal evidence. In most cases, CPS are implemented in conference rooms where environmental reviews are held. They are used as reference tools and are typically projected on a wall where participants may interact with them using infrared pointing devices.

CPS IMPLEMENTATION: WASHINGTON, D.C.

The Planning Support Systems group of MIT's Department of Urban Studies and Planning has recently completed an investigation of information use and communication in the planning organization. Using the National Capital Planning Commission (NCPC) as a test bed, the research group explored information management difficulties in the context of NCPC's planning process and how new technology and a sustainable information infrastructure could improve the land-use planning process. NCPC is the central planning agency for the Federal Government in the National Capital Region around Washington DC. This case describes a portion of the overall research effort that is concerned with supporting a collaboration between NCPC and the United States Department of Transportation (DOT) to study a corridor along Washington's North Capitol Street for a potential DOT headquarters site. This part of the research involved the development, testing, and initial implementation of a CPS.

CPS Structure

The CPS displays maps with various overlays that are linked to descriptive video images, sounds and text. It is implemented in NCPC's "commission room". The system uses a Macintosh Quadra 900 computer at the "front-end" of a heterogeneous network of IBM Compatible PCs and UNIX workstations at the NCPC. The Quadra, which runs Apple's Quicktime for digital video display and Aldus Supercard for media integration is hooked directly to a video projector and an infrared pointing device for user interaction.

In addition to the digital information contained in the "back-end" of the CPS, (GIS coverages, databases, images, etc.), a variety of physical information is routinely added to the system. This includes maps, images, text and video

tape that may have been collected as part of a planning review. This physical information is converted to a digital format using various peripheral devices such as scanners, video digitizing boards, and digitizing tablets. After this, the digital information can be transferred to the collaborative planning system, archived in the repository of digital information using the networked UNIX machines, or both.

There are two types of data communication between the CPS front and back ends, real-time and asynchronous communication. Real-time communication offers immediate responses to the CPS workstation resulting from queries sent to the UNIX network. This works best where information can be attained immediately through direct manipulation. Asynchronous communication is a means for displaying information that is not easily retrievable in real-time. An example of this type of communication is where a model that renders sunlight and shadow patterns may take several hours to generate a viable set of representations, several plausible scenarios can be modeled prior to a given meeting and then incorporated into a library of potential alternatives contained the CPS. The rapid display of this information can be attained using an appropriate interface metaphor, such as the hands of a clock, which would then yield a result.

Activities Supported by the CPS

The CPS supports several planning-related activities that focus on a selected site. These activities include: land use analyses, automobile traffic analyses, assessments of visual environments, and illustrations of proposed changes to the visual environments. These descriptions allow planners to begin the analysis of potential impacts by allowing them to rapidly retrieve, integrate and compare existing and proposed conditions. In this section we discuss several CPS components that make these activities possible.

Land Use Analysis The study of existing land use patterns is implemented through interactive land use, zoning, and building height maps. Each of these maps consists of transparent color polygons overlaid onto an aerial image of the study area. Summaries of the information represented by these polygons appear as the user selects the areas with a pointing device. As the user moves the pointer around the map, the displayed summary changes to reflect the zoning of the currently selected area. For the zoning and land use maps, the polygons and associated summary displays are color-coded to allow the user to visually distinguish them from other adjacent zoning classifications and land uses. In addition to the color coding, the zoning summaries contain a brief textual description of the zoning classification and information about maximum height, maximum lot coverage, and maximum floor area ratio for the selected area. The land use summary uses a color coded descriptor along with a textual descriptor derived from the draft environmental impact statement provided for the site.

Another description of land use is provided using a building height indicator. As the user points to a building on the map, the height of a selected building is indicated by a vertical bar in a window that also contains a representation of the U.S. Capitol building. When the pointer is moved around the map, the bar inside the window will slide up or down to represent the selected

building's height in relation to the Capitol building. The selected building's height (in feet) is represented numerically below the bar thus providing an example of multiple representations.

Automobile Traffic Analysis The study of existing automobile traffic flows uses an interactive traffic map that displays values of average daily traffic for selected street links in the study area. Traffic data, located in a window containing multiple representations of average daily traffic, is accessed by pointing to an associated street link on the interactive map. In addition to the traditional numeric representation, the average daily traffic values are represented graphically, with a bar; dynamically, with a clip of digital motion video; and audibly, with the level of traffic noise played back at the level experienced in the field. While the bar graph may look redundant in the static image portrayed in Figure 1, its utility becomes apparent as one points to different streets on the projected map, causing the bar to fluctuate. In this manner, users can compare relative levels of traffic to one another more easily.

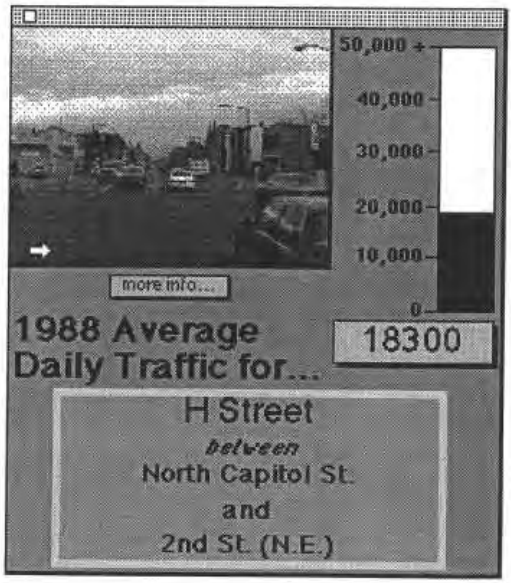


FIGURE 1: Average Daily Traffic Analysis

Assessments of Visual Environments To better understand an area under study, it may be necessary to view it from several different perspectives. The ability to analyze the existing visual environment in the study area is made possible through the use of digital video. Several types of digital video shots have been incorporated into the CPS. The three main shot types are fixed position, 360 degree axial view, and navigation.

The fixed position shots allow the user to view a video clip of a particular site from a fixed camera angle. They are symbolized on the visual quality map as

arrows that match the direction of the camera's angle. Some of the sites are viewed from several different angles that allow the user to "switch" perspectives the same way a television director can switch cameras to show different angles of a sporting event. In this manner, a subject can be viewed from several vantage points.

The 360 degree axial view allows the user to look completely around from a fixed vantage point. They are represented as circular symbols on the visual quality map. These views are useful for illustrating the environment surrounding a particular location. They allow the user to pan to the left or right by selecting the appropriate arrows at the bottom of the displayed image. The 360 degree axial views offer an additional view of what is behind the camera by allowing the user to pan around to look at the surrounding area.

The navigation images allow users to drive or fly through the study area. They are designed to aid visual navigation by enabling the user to view a geographic area from a moving perspective such as that experienced when traveling through a region. Navigation images are represented on the map as linear symbols that represent the routes available to the user. They are illustrated as large arrows in the lower right window of Figure 2.

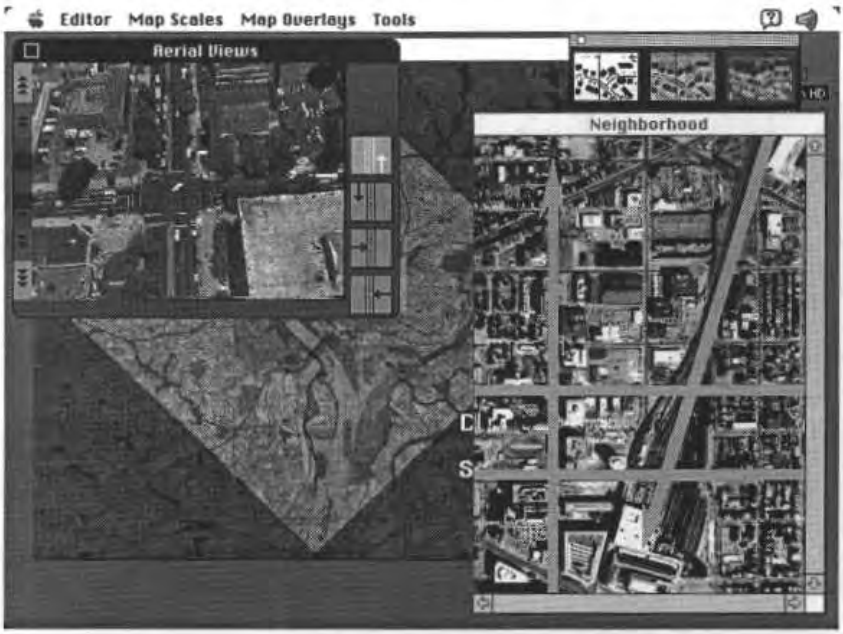


FIGURE 2: Visual Analysis using Aerial Images

The video image in the upper left corner of Figure 2 represents an oblique navigation image of a selected street from an altitude of 500 feet. The controller at the left edge of the "Aerial Views" window, (the upper left window), allows the user to control the direction of flight (forward or

reverse), as well as the speed of flight, by sliding the pointer towards either end of the controller. The user can determine the camera angle by selecting one of the iconic buttons at the right side of the "Aerial Views" window. The arrows on the icons represent the direction the camera was pointing with respect to the subject (in this case, the subject is the street).

Illustrations of Proposed Changes to Visual Environments The CPS enables users to qualitatively assess proposed changes to the visual environment by allowing easy access to images of architectural models and artists' renderings. By selecting the appropriate map overlay, a rendering of a proposed development appears at the appropriate geographic location on the base map along with arrows that are linked to various perspective views as shown in Figure 3. Selecting an arrow yields an image of an architectural model or rendering in a separate window, with controls allowing users to navigate around the image by "zooming" or "panning".

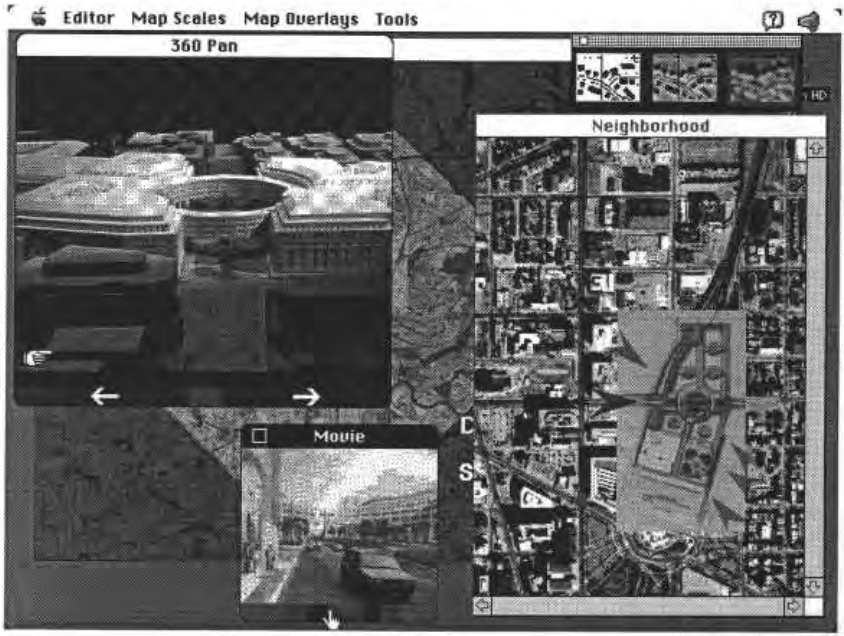


FIGURE 3: Illustrations of Proposed Changes to Visual Environments

These options allow users to inspect the proposed site in three ways. First, they can view the proposal from various perspectives around the site by selecting appropriate arrows linked to the map. Second, users can sequentially move through a series of site images while an associated arrow highlights on the map as each image is displayed. Finally, they can navigate around a specific rendering by zooming or panning with on-screen controls.

CPS Implementation

There are several specific contexts in which a CPS can be implemented. The system's wide range of usability makes it possible to adapt it to a variety of situations. Several of these situations are described below.

Demonstration of Capabilities Here the implementation of the tool serves to educate others about its power and capabilities. This education is also an important precursor to system implementation as a reference, data collection, or visualization tool. Often times, this is the only context in which the system is implemented.

User introduction to the system can be accomplished by relating the system to a familiar metaphor such as a slide show or a map. The next aspect of system implementation involves user familiarization with the system's geographic coverage area. This familiarization involves using system metaphors such as ground level images, aerial photos, and maps, to help users to orient themselves geographically to the community or study area. To familiarize users with the system's analytic tools, an example of an addressable problem can be worked through. In the context of such a problem, the users can familiarize themselves with the geographic coverage of the system and the analytical tools that the system contains. In this manner, the range of questions that the system can address can be defined for the users.

Presentation Tool In this context, the system can be used for the presentation of various proposals. The system's multimedia aspects make it a strong tool of persuasion. In particular, the ability of the system to use different media to represent different aspects of the same phenomena can enable a wide range of participants to reach a common understanding.

The systems persuasive abilities can also open the door to misuse. While it can clarify truths, it can also cast a veneer of legitimacy over untruths. This can be either an intended consequence or an unintended one. One can tell very different stories by presenting the same materials in different contexts. The use of representation aids can also aid the understanding of various models and scenarios. However, there is concern that representation aids can over simplify the output of a particular tool. This simplification can lead to a "black box" effect.

Facilitated Reference In this context the system acts as an information resource that can be drawn upon when needed. Information within the system can be attained 1) geographically, by participants pointing to a map, 2) visually, by participants scanning through a set of images while looking for specific visual criteria, and 3) textually, by searching for key words and phrases. These three avenues into the information will yield the display of phenomena using several media.

Data Visualization This context allows users to visualize large amounts of data such as traffic projections or shifts in demographics. This visualization is accomplished with the help of multiple representation aids. Representation aids can influence the display of various types of data so that they may be more readily understood by the users. This can be accomplished using multimedia tools such as digital video, animated graphics, and sound to make the human-machine interaction so engaging that the computer essentially becomes "transparent" to the human.

CONCLUSIONS

There is a considerable amount of work that is yet to be done in order to institutionalize this technology so that it can be routinely used in an organization without the need for extensive construction and reprogramming. On the software side, more effective tools need to be created that allow the rapid creation, organization, and linking of multimedia information by relative novices. Similarly, object-based programming tools need to mature to a point where they can be used by planning staff who are not "computer programmers". At the institutional level, there needs to be a commitment of staff time for training as well as the management of such a system. Additionally, there are several areas that beg further investigation as a result of this initial implementation. These include:

Analyzing public impact of the technology.

It is necessary to explore the benefits and drawbacks experienced when attempting to implement this technology in an environment intended to facilitate public discourse. The visualization tools contained in collaborative planning systems can empower groups and individuals who have traditionally been informationally disadvantaged due to a lack of technical sophistication, thereby allowing more people to become involved in the generation of alternatives. Exactly who benefits from such empowerment will depend on the situations in which the technology is implemented and begs further research.

Representation of information.

Just as these tools have the capacity to provide rich and compelling representations of environmental conditions, they have the capacity to provide rich and compelling misrepresentations. The issue of the relative trustworthiness of the information presented is going to float to the top of the discussions with increasing frequency.

Assessment of the cost-effectiveness of these tools.

While these systems have been demonstrated to be effective presentation aids, their cost effectiveness depends upon whether they are customized systems built entirely for a single 'presentation'; or generalizable, 'pre-fabricated' components that are easily assembled in Lego-like fashion from data, model, and graphics repositories.

Finally, the research opens the door to experimentation with more proactive planning styles using the new computing environment and information infrastructures. It is hoped that the planning process could be democratized using tools such as the Collaborative Planning System to bring data and analyses to a wider audience.

REFERENCES

Conklin, J. 1990. Hypertext: An Introduction and Survey, *IEEE Computer*, 20(9): 17-41.

Forester, J. 1982. Planning in the Face of Power. *Journal of the American Planning Association* 48: 67-80.

Hutchins, E. L., Hollan, J. D. & Norman, D. A. 1986. Direct manipulation interfaces. In Norman, D.A. and Draper, S.W., eds., *User Centered System Design: New Perspectives on Human Computer Interaction* (Hillsdale, NJ: Lawrence Erlbaum).

Klosterman, R.E. 1992. Evolving Views of Computer-Aided Planning. *Journal of Planning Literature* 6: 249-260.

Rasmussen, J. 1986. *Information Processing and Human Machine Interaction: An Approach to Cognitive Engineering.* New York: North Holland.

Shiffer, Michael J. 1992. Towards a Collaborative Planning System. *Environment and Planning B: Planning and Design*, 19: 709-722.

Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S., and Suchman, L. 1988. Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. In Greif, I. ed., *Computer Supported Cooperative Work* (San Mateo, CA.: Morgan Kaufmann).

Wiggins, L. L. & Shiffer, M.J. 1990. Planning with Hypermedia: Combining Text, Graphics, Sound, and Video, *Journal of the American Planning Association*, 56: 226-235.

Zachary, W. 1986. A Cognitively Based Functional Taxonomy of Decision Support Techniques. *Human-Computer Interaction*, 2: 25-63.

ACKNOWLEDGMENTS

Funding for this project was provided by NCPC contract # 91-02. As principal investigators to the overall project, Joe Ferreira and Lyna Wiggins helped to provide direction. Reg Griffith and Roy Spillenkothen provided unique executive insight, and Nyambi Nyambi and Enrique Vial coordinated implementation at NCPC. Additional support was provided by the MIT NCPC research team including John Evans, Rob Smyser, and Phil Thompson.

PROACTIVE GRAPHICS AND GIS: PROTOTYPE TOOLS FOR QUERY, MODELING AND DISPLAY

Barbara P. Buttenfield
NCGIA, Department of Geography, 105 Wilkeson,
SUNY-Buffalo, Buffalo NY 14261
internet geobabs@ubvms.cc.buffalo.edu

ABSTRACT

Recent developments in GIS hardware and software bring opportunities for creation of new tools for geographical research. Tools include multiple modes of presentation (multimedia) adding animation, sonification and video capture to static displays of text, tables, maps and images. These modes expand the available information channels, and empower viewer to incorporate conventional with experimental modes of data presentation. Hypermedia extends multimedia by linking the multiple channels of information transparently, enabling the viewer to choose the mode best suited to their preference and their application. Integrated with GIS techniques, hypermedia provides proactive control for steering computations, for spatial modeling, and for iconic query. Hypermedia tools effectively transform the user's role from what has been termed 'interactive graphics' into 'proactive graphics' (the term is proposed by this author).

This paper presents proactive tools for specific GIS operations, including database query (automated feature identification), modeling (calibration of a location-allocation model) and map display (automatic scale-changing and symbol modification). Implementations of proactive tools must consider both graphical principles (design and methods for evaluation) and computational issues (creation and maintenance of links between the prototype and the database or model). These issues will be discussed in the paper.

BACKGROUND

The growth of GIS as a scientific discipline has followed several prerequisite factors. The first relates to the commitment of U.S. national agencies in several countries to generate spatial data in digital form, and to place that information in the public domain. A second factor relates to software developments integrating spatial statistical models with GIS operations. A third factor relates to developments in data representation technology and techniques, including internal representations (improved data structures and algorithms for searching and sorting), and external representation techniques (advances in algorithms for data display). The research presented in this paper focuses upon visual tools that link internal and external data representations transparently.

The reliance in GIS upon visualization has for the most part been limited to external representation. This is ironic, given the reliance in GIS upon both display and upon analytical exploration. Both have strong traditions emphasizing the use of graphics to analyze data patterns, to generate and in some cases to test hypotheses (Cleveland, 1983). Given the current state of knowledge and current technology, visual tools in GIS can be implemented now to expand and refine analytical powers for exploration of geographical and statistical landscapes.

The Scientific Visualization initiative begun at the National Science Foundation (McCormick et al, 1987) marks a chronological point at which the scientific community formally recognized the potential for visual tools to be integrated into numeric modeling and analysis. The McCormick Report defines visualization as

"... a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. ... Visualization embraces both image understanding and image synthesis. That is, visualization is a tool both for interpreting image data in a computer, and for generating images from complex multi-dimensional data sets." Implicit in these statements is a belief that visual displays can be used to analyze as well as to illustrate spatial information, to generate hypotheses as well as interpret scientific results. At the time the McCormick Report was published, the advent of low-cost graphics software was substantially changing methods of data representation. Exploratory Data Analysis (EDA) techniques developed by Tukey and colleagues at Bell Labs during the past decade have a strong graphical component, and their incorporation into statistical packages effectively tied graphics to statistical description and exploration. Demonstrations of individual techniques such as data brushing have begun to appear in the cartographic literature (MacDougall, 1992; Monmonier, 1992).

In GIS interactive computing environments, then as now, users are presented with map displays based upon graphical defaults that rarely ascribe to sound principles of cartographic design (Weibel and Buttenfield, 1992). In spite of the fact that GIS systems rely heavily on map displays of archived data, data manipulation and map manipulation remain isolated tasks. For the most part, the dialog between system and user remains constrained by a limited number of commands pre-defined by system developers. From a human factors point of view, it seems likely that requiring that users type keyboard commands to modify maps and screen displays reduces efficiency for the user, and may increase fatigue levels and propagate 'use error' (Beard, 1989). This paper argues for refinement of current interactive graphics modes to adopt 'proactive' tools that will extend and facilitate GIS users capabilities.

CURRENT NEED FOR IMPROVING VISUAL TOOLS

Capabilities are at hand now to improve visualization tools in GIS environments. Currently available tools incorporate multiple modes for presenting information, including text and graphics and animation, sonification and video capture. Multimedia presentations are intended to expand the information processing channels available to viewers. Examples of geographic multimedia applications include electronic atlases such as produced for the State of Arkansas by Richard Smith, and Rhind et al's (1988) Domesday Machine.

Hypermedia is a special form of multimedia, in which documents contain a collation of disparate media forms presented on a single presentation device, typically a computer. Hypermedia extends multimedia by linking the multiple channels of information transparently. That is, users can select a mode of information presentation they prefer, and can make logical leaps in a thread of database query that may not have been anticipated by system designers (Barker and Tucker, 1990 p. 20). Hypermedia provides an excellent example of proactive software functionality.

Chronology of Hypermedia Development

In 1945, President Roosevelt's Science Advisor developed the concept of a "memex", a tool for linking any two items in large collections of information (Bush, 1945). Bush (1945, p. 106) stated that the human mind operates most efficiently by association and 'logical leaps' as opposed to following a single logical sequence. Systems offering query by association are considered easier to learn and to use in the context of the natural structure of human memory proposed by current semantic network models (Lachman et al, 1979; Smith and Weiss, 1988).

Hypertext has remained the most commonly encountered form of hypermedia. Engelbart (1963) developed the NLS (oNLine System) at Stanford that incorporated hyperlinks into a text database. Other hypertext systems have included Nelson's XANADU system and Van Dam's Hypertext Editing System (Conklin, 1987). Brown University's Intermedia Project exemplifies more recent systems extending hypertext capabilities by incorporation of links with graphics displays. Apple Computer's HyperCard package, marketed in the last decade, provided a first attempt to merge hypertext functions with multimedia including graphics, crude animation, and sound. Additionally, HyperCard included a scripting language (HyperTalk) for linking text, graphics and sound using an index card metaphor. Scripting capabilities are fundamental to proactive graphics: GIS users should be able to script the paths they choose to follow for internal and external data representation. To date, these capabilities are not readily available.

Following and expanding upon HyperCard, other scripting languages have been developed, including Kaleida SCRIPT-X developed jointly by Apple Computer and Toshiba Corporation. The Macintosh product Macromedia Director combines animation functions, a graphics editing module and a hypermedia scripting language (LINGO) with sound channels and video capture; this product and this platform continue to be the choice for many professional multimedia and hypermedia producers. Cartographic adoption of animation authoring tools has begun to reappear in the literature after a 10-15 year moratorium (Weber and Buttenfield, 1993; MacEachren and DiBiase, 1991; Gersmehl, 1990), although to date the animations are illustrative, and do not incorporate proactive functions. One exception to this is a hypertext document under development at NCGIA-Buffalo (Buttenfield and Skupin, 1993) designed as an online browser for federal digital data product specifications. In its current version, scripting capabilities are exploited, but are not yet available to users. This means that while an infinite number of paths might be traversed through the Browser, users must reconstruct their path each time they browse -- they cannot 'script' a particular path to follow in future browsing, nor document where in the browser they have traveled so far. The author's vision for fully operational proactive graphics incorporates these functions as minimal system requirements.

Applications of Hypermedia to Information Systems

Examples of hypermedia systems in current use are widespread. At the National Gallery in London, a hypermedia tourguide system gives visitors access to museum holdings, with online images of paintings arranged by artist, by nation, or by century. Visitors can select the artwork they wish to see and request a hardcopy map of the museum guiding them to the art they have selected. In other domains, hypermedia packages are used in large office complexes to configure heating/cooling systems, and to install and maintain integrated software, such as the VAX VMS operating systems (Horton, 1991). Hypermedia systems maintain links across distributed networks, such as 'anonymous ftp' and some online bulletin board systems (Martin, 1990). In every case, the information at issue exists in multiple formats, and decision support is based on associating a full range of information presentation modes.

It seems obvious to apply hypermedia to the interrelated data types common in GIS and Spatial Decision Support Systems (SDSS). Multiple modes of information characterize the nature of geographical data. Tabular, numeric and cartographic information each serve an important role in GIS query, analysis, and display. To date, system designers have not incorporated multimedia and hypermedia methods in GIS functionality. Hypermedia may empower viewer flexibility by incorporating conventional with experimental modes of data presentation, enabling the viewer to choose. Hypermedia links in GIS databases may remove constraints to explore information in a predetermined sequence or by

isolated commands. Integrated with GIS techniques, hypermedia should provide proactive command for steering computations, interactive modeling, and queries.

The added challenge for GIS functions is the integration of internal and external data representation, discussed above. GIS users should be able to manipulate external representations (map or chart displays) to modify internal representations (database contents and pointers). Human acuity for visual pattern recognition lends itself readily to understanding spatial patterns. Much of GIS user activity involves making maps, changing maps, and compositing maps. The continuing insistence of GIS system designers that users must learn some arcane syntax of typed commands to perform map operations such as buffering and overlay is incomprehensible, given current state of knowledge and technology in other scientific disciplines. Why should users be forced to separate spatial data manipulation from map manipulation? This is the driving force justifying adoption of proactive graphics, and the major distinction between hypermedia tools and proactive tools.

Applications of proactive tools in GIS that appear both implementable and beneficial (in the sense of enabling better access and use of information) include four areas of activity.

- For **spatial query**, which in most GIS systems revolves around combination of Boolean text strings, a hypermedia system might allow proactive selection ('point-and-click') of map objects or spreadsheet data objects to be expanded upon, literally, numerically, or iconically.

- For **spatial modeling**, hypermedia tools may provide proactive modification of parameters, and alert researchers to drifting values in sensitivity analysis, optimizing for equity or efficiency according to user choice.

- For **cartometric analysis and display**, hypermedia tools might steer computations, or demonstrate the impact of a particular algorithm in a particular domain.

- For **online documentation**, hypermedia systems exist now, in the form of online help systems and online system manuals. Personal computer users are commonly familiar online help files accompanying workstation software (e.g.., UNIX user's manual, Macintosh and DOS-Windows' WORD).

PROACTIVE GRAPHICAL TOOLS

What is needed to fully integrate the user into the GIS process is provision of visual tools enabling **proactive** user involvement, as opposed to **interactive** involvement. There is a need for graphics for view by people whose knowledge of a particular dataset is deep, and whose interest in developing cartographic or system expertise is overshadowed by an interest in an immediate application or domain of inquiry. These include planners, natural resource managers, and social service providers, to give some examples. In some use situations (e.g.., environmental emergency planning), the time available to become facile with system use is quite low, in contrast to the level of training required to learn most GIS command languages.

What is Proactive Visualization?

The term **proactive** is used here in its conventional sense, that refers to taking action before it is requested or mandated. Its connotation is assertive. The

prefix 'pro-' is taken from the Greek **pro** meaning "prior" or "before". Interactive computing provides capabilities to respond to system dialog boxes and menus, and limited capabilities to take actions anticipated by system designers (e.g.., opening and saving files in pre-determined formats, generating displays according to system graphic defaults). Proactive computing, also referred to 'hyper-active' computing, (Laurini and Thompson, 1992) simulates a system responsive to commands and queries that may not have been anticipated by system designers. A fully operational proactive computing environment incorporates a scripting language that allows users to implement their own commands. In GIS environments, where so much of the information stream is visual, proactive computing becomes nearly synonymous with proactive visualization, where users actively manipulate images and icons to enable system commands or database queries, and to steer modeling and computation (Parsaye et al, 1989).

Examples of Proactive GIS Tools

The first example demonstrates proactive query of map features for automatic scale changing, involving real-time coordinate simplification with viewer control. A second example demonstrates automatic feature identification. A third example simulates a location allocation model, with real-time selection of candidate facilities coupled with system adjustment of routes and additive capacities. These are not a comprehensive taxonomy, but they do provide examples of query, analysis and display in a GIS context.

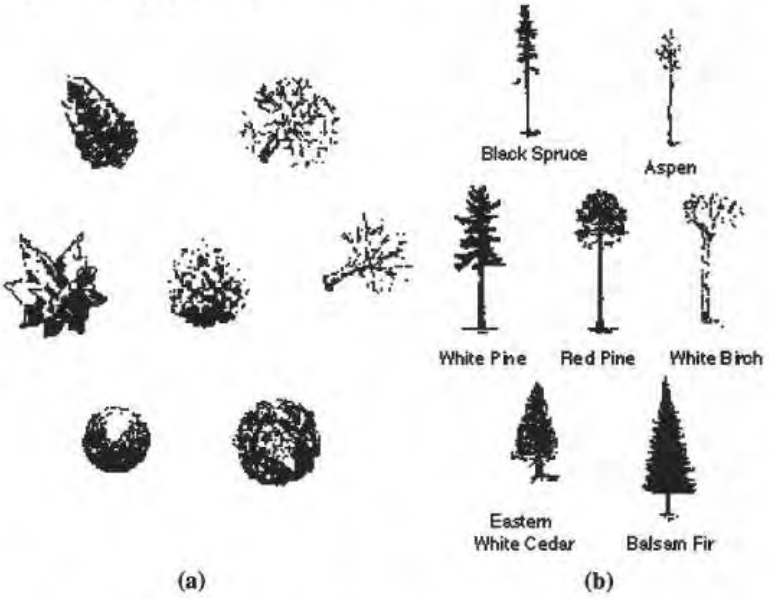


Figure 1
Photointerpretation keys often include iconic look-up tables matching (a) overhead views with image displacement with (b) silhouettes identifying feature shadows on flat terrain. (Taken from Lillesand and Keifer, 1979: 132-133).

Automatic Feature Recognition (spatial query). Spatial query in most GIS systems involves forming combinations of Boolean text strings to search and select items from a database. Often the results of a query are displayed

graphically. Tomlin (1990) refers to GIS query and display collectively as "cartographic modeling". In a remote sensing application, a prototype might take the form of a photointerpretation look-up key (Avery and Berlin, 1992; Lillesand and Kiefer, 1979). The purpose of such keys is to identify features in an aerial photograph, where images are sometimes distorted or cast in shadow, or partially obscured by adjacent objects. Keys may provide verbal, numeric or iconic information. When photointerpretation keys are iconic, they often illustrate the appearance of a feature in displaced form, as on an unrectified photograph. Alternatives to this include keys illustrating the shadow cast by a feature upon flat terrain. The proactive tool in this type of application (Figure 1) would respond to a windowing or lasso command by searching a look-up table for possible pattern matches and displaying each of these. An expert system implementation could attach confidence values to each key in accordance with user-supplied information about the site and situation of the photograph.

Location Allocation Modeling (spatial modeling; example also reported in Buttenfield and Weber, 1993). The customary sequence of actions in location modeling is to input locations and capacities for nodes from keyboard and/or digitizer, and to allocate nodes to central facilities by attribute entry. Selection and modification of parameters are accomplished in batch mode; modeling runs are not directly linked to the GIS system nor to its associated database. Visualization tools are limited to illustration of solutions (Armstrong et al, 1992).

Proactive graphics may assist with model preparation and analysis, as for school bus allocations. (Figure 2). In the figure, the diamond represents a school whose district is experiencing population growth. Circles represent existing bus drop points, with shading representing the capacity of students at each location. The problem of adding additional drop points can be solved by pointing to locate a new drop point, and dragging a straight line (shown in thick black line) over to the school to be served. A dialog box allows the user to enter capacity figures. The system response is displayed in the right side of the figure.

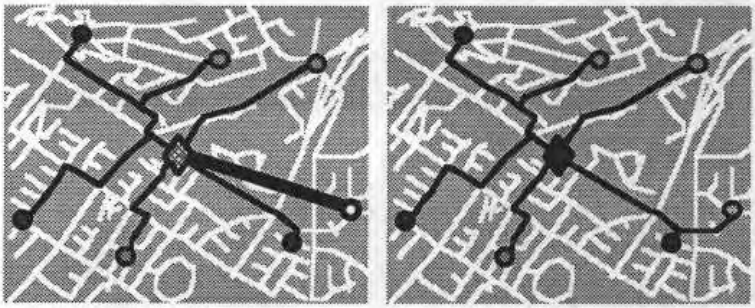


Figure 2
A Proactive Tool for Location Allocation Modeling

The allocation model selects a logical street path for the bus to follow, accommodating problems presented with the limited access highway by selecting a meaningful crossing point. Additionally, the system adjusts additive capacities at the selected drop point as well as at the intermediate points. The prototype simulates this procedure, providing a number of facility placement options that will snap into place depending on where the user points, and then provides an appropriate solution image and numeric tabulation. The number of options will depend on available computer memory, as a different solution will need to be kept in memory for each selectable option.

Automatic scale-changing (cartometric analysis and display). It is possible to model map features, by simplification of coastline detail, aggregation of buildings or small lakes, and similar operators that reduce detail. This type of transformation belongs in the realm of display, as it is the view on the database (not the database itself) that is being modified.

The prototype simulates an application where a user must decide how detailed the representation of base map features should be for a given application. The decision involves query of coordinate strings and processing by a simplification or smoothing routine (commonly the default routine is from Douglas and Peucker, 1973). Figure 3 shows a portion of Norway's Sonnenfjord. Imagine that the user has selected this portion from the rest of the coastline using a lasso command; the proactive tool responds by placing this coastline segment into a small tear-off window floating above the base map. The user can make decisions as to how much to simplify the final base map by operating on the windowed portion. Working with a subset of the coordinate file makes it possible for the CPU to generate real-time feedback. Proactive simplification may be guided by means of a slider bar. The resulting generalized coastline is not a simulation, but a real-time computation that the users commands proactively.

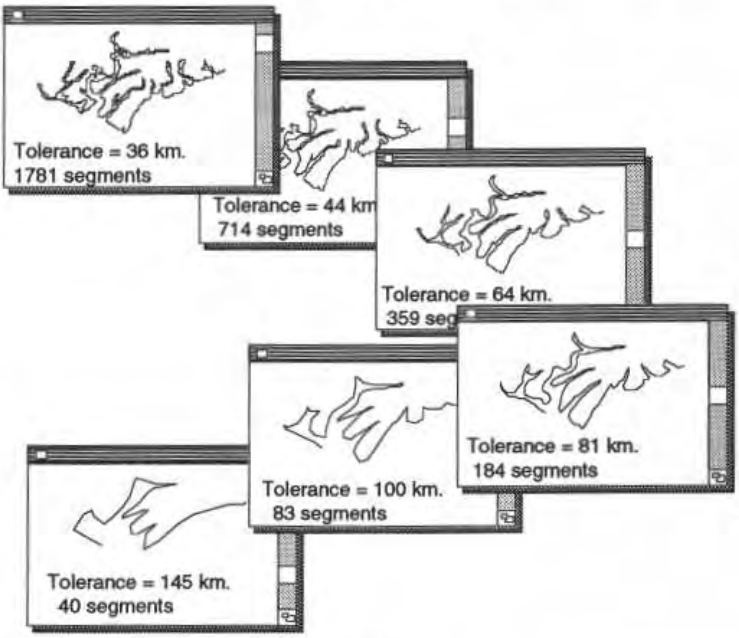


Figure 3
A Proactive Tool for Automatic Scale Changing

This is not a zoom-in with pixel replication, but a real-time query with coordinate simplification. As the user slides the scroll bar downward, the system simplifies the windowed subset of coordinates, replacing the view in the window in real-time and providing descriptive statistics (average tolerance of retained details, in kilometers, and total number of segments in this window). Once the user determines a tolerance value that looks appropriate, the system can be commanded to simplify the rest of the base map representation. Proactive visualization allows the user to see beforehand what the impact of tolerance value selection might be, without possibly wasting CPU processor cycles only to

discover the base map is too simplified, or too complex, for the application. Variations on the prototype development include selecting a different algorithm, changing the tolerance threshold, and choosing a specific window within which the chosen simplification model and parameters should be applied.

SUMMARY

Recent developments in Geographic Information Systems (GIS) hardware and software bring opportunities for the creation of new tools for geographical research. Cartographic tools include multiple modes of presentation such as animation, sonification and video capture. Multimedia presentations are intended to expand the channels available to viewers for information processing. Hypermedia extends multimedia by linking the multiple channels of information transparently, effectively transforming the user's role from what has been termed 'interactive graphics' into what could be called 'proactive graphics'.

This paper reports on three prototype applications, demonstrating proactive spatial modeling, iconic queries, and automatic scale-changing. To date, system designers have not incorporated proactive methods in GIS functionality. Proactive tools empower viewer flexibility by incorporating conventional with experimental modes of data presentation. Proactive links in GIS databases may remove constraints to explore information in a predetermined sequence, to interact with spatial data by manipulating maps and images, in addition to or in place of conventional database SQL commands. Integrated with GIS techniques, proactive tools should provide user control for internal and external data representations.

ACKNOWLEDGMENTS

This research forms a portion of NCGIA Research Initiative 8, "Formalizing Cartographic Knowledge", funded by the National Center for Geographic Information and Analysis (NCGIA). Support by the National Science Foundation (NSF grant SES 88- -10917) is gratefully acknowleged.

BIBLIOGRAPHY

Armstrong, M.P., Densham, P.J., Lolonis, P., and Rushton, G. 1992 Cartographic Displays to Support Locational Decision-Making. **Cartography and GIS**, vol. 19(3): 154-164.

Avery, T.E. adn Berlin, G.L. 1992 **Fundamentals of Remote Sensing and Airphoto Interpretation**. New York: Macmillan (5th Edition).

Barker, J. and Tucker, R. N. 1990 **The Interactive Learning Revolution**. New York: Kogan Page, London/Nichols Publishing.

Beard, M.K. 1989 Use Error: The Neglected Error Component. **Proceedings**, AUTO-CARTO 9, Baltimore, Maryland, March, 1989: 808-817.

Bush, V. 1945 As We May Think. **Atlantic Monthly**, No. 7:101-108.

Buttenfield, B.P. and Skupin, A. 1993 **SDTS Browser.** Buffalo, NY: NCGIA-Buffalo, 301 Wilkeson, SUNY-Buffalo.

Buttenfield, B.P. and Weber, C.R. 1993 Visualisation and Hypermedia in GIS. In Medyckyj-Scott, D. and H. Hearnshaw, (Eds.) **Human Factors in Geographic Information Systems**. London: Belhaven Press (forthcoming).

Conklin, J. 1987 Hypertext: An Introduction and Survey. **IEEE Computer**, vol. 2(9): 17-41.

Douglas D H and Peucker T K 1973 Algorithms for the reduction of the number of points required to represent a line or its caricature. **The Canadian Cartographer** vol. 10(2): 112-123.

Engelbart, D. 1963 A Conceptual Framework for the Augmentation of Man's Intellect. In Howerton, P.W. and Weeks, D.C. (Eds.) **Vistas in Information Handling**. Washington DC: Spartan Books, vol.1: 1-29.

Gersmehl, P. J. 1990 Choosing Tools: Nine Metaphors of Four-Dimensional Cartography. **Cartographic Perspectives**, vol. 5: 3-17.

Horton, W. 1991 **Illustrating Computer Documentation**. New York: Wiley.

Lachman, R. Lachman, J.L. and Butterfield, E.C. 1979 **Cognitive Psychology and Information Processing : An Introduction**. New York : Halsted Press.

Laurini R. and Thompson, D. 1992 **Fundamentals of Spatial Information Systems**. London: Academic Press. (See especially Chapter 16 on Hypermedia).

Lillesand, T. M. and Kiefer, R. W. 1979 **Remote Sensing and Image Interpretation**. New York: Wiley.

MacDougall, E. B. 1992 Exploratory Analysis, Dynamic Statistical Visualization and Geographic Information Systems. **Cartography and GIS**, vol.19(4): 237-246.

MacEachren, A.M. and D.W. DiBiase 1991 Automated Maps of Aggregate Data: Conceptual and Practical Problems. **Cartography and GIS,** vol. 18(4): 221-229.

Martin, J. 1990 **Hyperdocuments and How to Create Them**. N.J.: Prentice-Hall.

McCormick, B.H., DeFanti, T.A., and Brown, M.D. Eds. (1987) Visualization in Scientific Computing. **Computer Graphics** vol. 21(6) (entire issue).

Monmonier, M. S. 1992 Authoring Graphic Scripts: Experience and Principles **Cartography and GIS** vol. 19(4): 247-260.

Parsaye, K., Chignell, M., Khosshafian, S. and Wong, H. 1989 **Intelligent Databases: Object-Oriented, Deductive, Hypermedia Technologies.** New York: John Wiley and Sons, Inc.

Rhind, D., Armstrong, P. and Openshaw, S. 1988 The Domesday Machine: A Nationwide Geographical Information System. **The Geographical Journal**, vol. 154: 56-68.

Smith, J.B and Weiss, S.F. (Eds.) 1988 Hypertext. **Communications of Association for Computing Machinery**, vol. 31(7), entire issue.

Tomlin, C.D. 1990 **Cartographic Modeling**. Englewood NJ: Prentice-Hall.

Weber, C.R. and Buttenfield, B.P. 1993 A Cartographic Animation of Average Yearly Surface Temperatures for the 48 Contiguous United States: 1897-1986. **Cartography and GIS**, vol. 20(3): 141-150.

Weibel, W.R. and Buttenfield, B.P 1992 Improvement of GIS Graphics for Analysis and Decision-Making. **International Journal of Geographic Information and Analysis**, vol. 6(3): 223-245.

A SPATIAL-OBJECT LEVEL ORGANIZATION OF TRANSFORMATIONS FOR CARTOGRAPHIC GENERALIZATION

Robert McMaster **Leone Barnett**
Department of Geography **Department of Computer Science**
University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

A current research thrust in geographic information systems is the attempt to add generalization functionality to existing software. However, it is difficult to create user-friendly environments for non-experts attempting to generalize maps because the organization and naming (or description) of generalization operators is based on a mixture of human-focused functionality and computer-focused data structuring. The generalization operators have names such as simplification, smoothing, displacement, enhancement, typification, and exaggeration, which suggest their functions. However, they are commonly divided into raster and vector groups of operators, which specify their applicability to a data structure or their geometry (points, lines, and areas). Analysis of these operators, focused on an examination of their higher level, or human oriented, purposes, reveals that there is redundancy among them, and, at times, multiple purposes embedded within them. In this paper, a structure for generalization operations is created that addresses the different levels of abstraction inherent in understanding the operators. While most existing frameworks organize operators at the data structure or conceptual raster or vector level, the spatial-object level may be used for a higher-level organization. Such a structure offers an alternative to designing and implementing generalization operators based solely on the raster/vector dichotomy and geometry.

INTRODUCTION

Current methods of automated cartographic generalization, although useful and necessary for manipulation of geometric objects, fall short of providing a system with general capabilities that produce adequate results. There is a growing awareness that cartographic features are not simply geometric objects. According to Muller (Muller 1991), "more sophisticated methods will be required which take into account the phenomenal aspects of a line." This will include system use of rules that allow more intelligent decision making.

At this point, there is not even a clear set of concepts concerning generalization that establish standards for the design of a generally useful interface, allowing a human to interact with the system at a relatively high level. Currently, humans must be very tuned in to the details of the geometric representations of their data (i.e. vector, raster, point, line, etc.). This is not likely to be the aspect of their data that is of utmost importance to them, but they are forced to consider this aspect to use the system effectively.

The problem of how to incorporate real-world, or "phenomenal," information in a manner that can facilitate automated generalization can be approached from a perspective of "semantic information" analysis. In this context, semantic information is whatever information is necessary to understand the meaning (or meanings) of data stored in the computer. System support of this information means the system has the ability to attach appropriate meaning (via other data and operations) to the data it controls. Analysis, in this case, of generalization operators, is necessary because in order to codify and represent the meaning of the data the operators are applied to, we need a better understanding of what the operators actually do.

To facilitate this analysis, we offer a perspective on the relationship between geometric or spatial data, and real-world data. This perspective is summarized in Figure 1. Using

this, we can understand and classify generalization operators at a spatial level of abstraction that removes the concern with vector and raster representations at the higher level, while maintaining them, appropriately, at the lower level. Before we discuss Figure 1 as the basis for a new, semantically oriented, approach for organizing generalization operators, we review previous approaches to operator classification.

PREVIOUS WORK

Some authors have attempted to formulate rules concerning generalization. In trying to understand what rules could guide the generalization process, authors have come up with different ways to think about and describe the different generalization tasks involved. They organize the tasks in terms of immediate purpose, high level function, focus of use, and/or constraints.

We begin with Brassel (1985) who developed one of earliest comprehensive listings of operations. Brassel organized generalization operations by geometry, and included classes of (1) point features, (2) line features, (3) area features, and (4) volume features. The point features category included operations such as expand, displace, classify, and display while several of those operations found in the area features category included select, eliminate, classify and displace. Brassel's structure included redundancy among the four categories, as well as operations that many authors would not include as part of generalization (such as the line features operation of "change topology in linear networks"). It was, nonetheless, the first attempt to organize what had been, up to that time, a rather haphazard development of generalization operations.

Beard examined generalization operators with respect to the formalization of generalization rules. Beard's view of generalization operations is based on the constraints that are pertinent to generalization. The four constraints include (1) graphic, (2) structural, (3) application, and (4) procedural. Procedural constraints, for instance, control the order and interaction of actions or operations as well as the order in which these constraints are satisfied (Beard, 1991). As shown below, Beard identifies seven operations for generalization. She categorizes them into three broad classes, including: (1) operations to reduce the number of objects, (2) operations for simplifying the spatial domain, and (3) operations for simplifying the attribute domain. Those operations designed to reduce the number of objects, for instance, include select, group, link, and aggregate. Those operations designed for simplifying the spatial domain include link, aggregate, collapse, and simplify. Classification is designed for simplifying the attribute domain. Each of these operators is then categorized in terms of graphic, structural, and application constraints.

Generalization operations: Beard

(1) select
(2) simplify
(3) collapse
(4) link
(5) aggregate
(6) classify
(7) group

Generalization techniques: Mackaness

(1) change symbols
(2) mask symbols
(3) increase size difference
(4) select
(5) omit
(6) simplify
(7) combine (reclassify)
(8) displace
(9) exaggerate

Mackaness (1991) provides an alternative approach. In identifying nine generalization techniques, he focuses on the process and the structure of the processes. Some of these techniques, such as select, omit, simplify, combine, displace, and exaggerate are similar to those of other investigators. Others, including change symbols, mask symbols, and increase size difference, relate more to the manipulation of symbols than to geometric generalization. Mackaness proposes that both rose diagrams and thermometers can be used to measure the interaction among and individual performance of the operations.

The structure proposed by McMaster and Monmonier, and reported in McMaster (1991), classifies the operators of generalization into those designed for manipulating the geometric description of geographical elements, and those designed for manipulating the attributes of (typically, simple cell) elements. Operators involving attribution include classification and symbolization. Many authors consider classification, symbolization, and simplification to be core concepts of map generalization. A further division of operations identifies those used for raster and vector data structures. For instance, generalization of raster data can involve structural, numerical, numerical categorization, and categorical classes of operations. Generalization of vector data involves operators that are designed for a specific vector data type--point, line, area, and volume. Commonly, those operators applied to linear features in vector format include simplify, smooth, displace, merge, enhance, and omit.

Despite the reasonably successful attempt to clearly distinguish between vector and raster operations, a structure such as the one described above, ignores the fact that redundancy exists both between raster and vector operators, and among operators for different vector data types. Additionally, such a structure would not facilitate the use of semantic-level information in improving the results of generalization. Most of the operations, and specific algorithms, have been designed at the data structure level or the conceptual raster or vector level.

It is our goal to understand and organize generalization operators, in terms of the functional tasks they actually perform, in a manner that makes clear when differences are important, and when similarities should be acknowledged. To this end, we have developed a new model, based on levels of meaning, that provides a context for understanding and specifying the meaning of operators.

OVERVIEW OF LEVELS

In Figure 1 we present a model of how different types of information may be organized into levels of "meaning." The "lowest" level, shown at the bottom, is called the "Data Structure or Implementation Level." The "highest" level, shown at the top, is the "Real-world Phenomena Model Level." This level might also be called the "Entity Level" by those familiar with the Spatial Data Transfer Standard (SDTS) terminology (National Institute of Standards and Technology 1992). Referring to a level as "high" or "low" is somewhat arbitrary. However it does reflect that concepts primarily of interest to humans are called "high level" relative to "low level" concepts primarily useful in computer programming. The levels that involve real-world information are also levels where many types of "semantic" information may be found.

The model was developed as a means of understanding the generalization operators, both existing and desired, in terms of functionality. For our purposes, functionality refers to specific types of data transformation and manipulation. A justification for modeling the different levels of meaning that are involved in generalization is, based on what level you assume an operator exists, the meaning of the operator (its function) is different. The following three assumptions concerning generalization are the basis for the structure of the model:

1. There are two traditional, and very different, classes of data transformations that occur as part of cartographic generalization. One is spatial-object based (i.e. based on geometry), and the other is real-world information based. (Often, real-world information is stored as attributes for raster data.)
2. The term "attribute" is over-loaded, and depending on what level of meaning is assumed, an attribute plays a different role and takes on a different meaning.
3. A conceptual spatial-object level subsumes lower level spatial models such as those known as vector and raster. There are limitations or constraints on operations that may be fundamental to a lower level model, such as vector or raster, but do not apply at the conceptual spatial-object level.

Note that the assumptions as stated go slightly beyond the commonly accepted view that generalization predominantly involves spatial transformations and attribute transformations. While this is true, it doesn't explicitly emphasize the important connection between attributes and real-world information. In addition, attributes may be any type of data, including spatial or geometric. Assumption number 1 above more clearly emphasizes that any spatial transformation is in a separate class from transformations requiring some sort of real-world data.

In the next section, each level is described in more detail. We identify the major classes of activity or function in each. We discuss how some existing generalization operators fit into these classes, and that some even fit into multiple classes because they serve multiple functions.

DESCRIPTION OF LEVELS

Data Structure or Implementation Level

The "Data Structure or Implementation Level" is of the least importance in this paper, but is included to illustrate the difference between a *conceptual* vector or raster structure and its actual *implementation*. This difference exists since a line may be represented many ways. For example, a line at the conceptual vector level, may be implemented at the lower level, as a sequence of x, y coordinate pairs, with each list stored in a separate record. Alternatively, the same conceptual line may be represented as a line object, where its line-id either points to or is related to a separate file containing points. This level may be broken down further, if desired, as was described by Peuquet (1990). However this is the lowest level we need to distinguish for the purposes of this paper. At this level, attributes are any values, irrespective of meaning, that are associated with the data structures used for either the vector or raster objects.

Conceptual Vector or Raster Level

This level ideally could include all lower level spatial models that manage various types or aspects of spatial data. For now, due to the types of operators developed for generalization, this level comprises two separate models that have received much attention in the literature. These separate models, often referred to as the vector and raster models, are sometimes described as the "dual" of each other (Peuquet 1988). The conceptual raster model is based on a regular partitioning or tessellation of space. The conceptual vector model is based on a point, line, and polygon description of the geometry of an object in space. Attributes at this level are, again, any values, irrespective of meaning, that are associated with the vector or raster primitive objects. These include points, lines, and polygons for the vector model, and raster cells for the raster model. This level is significant since many operators have been developed with this level in mind, as described previously.

In subsequent sections we examine transformations that relate specifically to either the conceptual raster model or the conceptual vector model. We identify general categories of activity for each of these, and we indicate which generalization operators exemplify an activity, at least as far as one of their functions is concerned. The conceptual vector model is discussed next. Then, just prior to discussing the conceptual raster model, some higher levels are introduced.

Conceptual Vector Level

Transformations that relate to the conceptual vector model are easily confused with those of the next higher level, which is called the "Conceptual Spatial Object Level." This is because the higher level subsumes the conceptual vector model. This means that many spatial transformations can occur at both of the levels involved. The major distinction between the conceptual spatial-object level and the conceptual vector (or raster) level centers around the definition of an area. At the conceptual vector level, the basic objects, or primitives, include the point, the line, and the polygon. The line is a

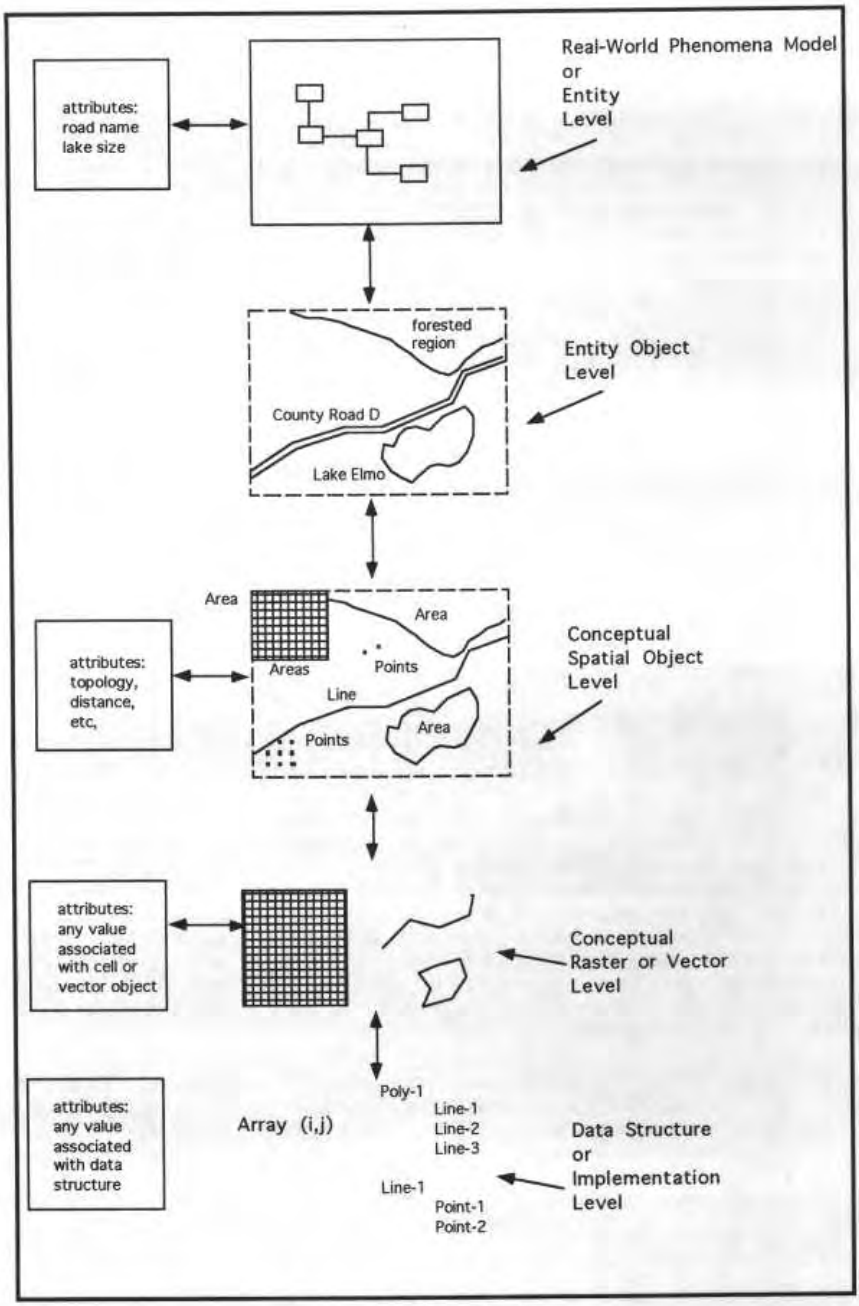


Figure 1: Levels of Meaning

sequence of points, and the polygon may be either a closed sequence of lines or points. The polygon defines the boundary of an area, but that is the only way in which an area may be represented. At the conceptual spatial-object level, the primitives are the point, the line, and the area. An area in this case is a more complicated notion. It includes the

type of area commonly represented by a raster cell. It also includes the type of area that may be represented by a cluster of contiguous raster cells. Thus at the conceptual spatial-object level, the area concept includes, but is not limited by, the conceptual vector means of modeling area.

This is a subtle distinction, and often one that is either not explicitly recognized, or is ignored in the discussion of generalization operators. The result is a use of terminology that is not quite clear. For example, a discussion of the "smoothing" operation, without specification of the level of interest, may refer to vector line smoothing, or a raster smoothing operation called "erode smooth," or smoothing of any spatial object that happens to be a line. In fact, an author may have the conceptual spatial-object level "in mind," yet discuss the concepts as if the conceptual vector-level constraints must apply.

Given the conceptual vector primitives, the conceptual vector-level transformations mostly concern conversions from one type of primitive to another of the same or different type. Many of these conversions are identified and named (as generalization operators) by McMaster and Shea (1992). Their names of groups of vector generalization operators include "Point feature generalization," "Line feature generalization," and "Area feature generalization." Their first group includes "aggregation" and "displacement." Their second includes "simplification," "smoothing," "displacement," "merging," and "enhancement." The last one above includes "amalgamation," "collapse," and "displacement."

This is not a false categorization for the conceptual vector-level operators, but it does not serve to completely and precisely specify all such operators. For example, the same name is used to specify more than one operation (as in "displacement") making that operator imprecise at the vector-model level. "Aggregation," which operates on points, may indicate a points-to-area transformation or a points-to-point operation. If it specifically means the points-to-area operation, what operator handles points-to-point? An operator that is missing is one that involves the removal of a common boundary line between two areas (i.e. polygons). Although similar to "amalgamate" it is technically different, since one involves contiguous polygons with a common line boundary, and the other involves non-contiguous polygons.

Given the lack of fine distinctions of these operators with respect to the conceptual vector level, it may be that the operator names are more well suited to the conceptual spatial level instead. Indeed, that is the level some authors intend when they use these operator names. That may be reasonable to do, but there is clearly a difference in the understanding of what the operator does if the context is one level or the other.

A more complete organization of conceptual vector-level transformations would specify, within each of the broader categories of point, line, and polygon generalizations, exactly what geometric changes occur. The specifications would include what type the original object would change into (e.g. points-to-area), as well as how many of the original objects are to be generalized (e.g. one vs. more than one). In addition, the contiguous or non-contiguous arrangement of the objects should be factored in. As an example, an operation such as line simplification could be classified as a 1-line-to-1-line type of transformation, and aggregation as a many-points-to-1-polygon transformation. If a many-points-to-1-point operation is desired (which some may view as an aggregation operation also), the difference between the two kinds of aggregation can be stated clearly and better understood. Although at a higher conceptual level the distinction may not be of interest, at the lower levels, it is important. The design of algorithms to perform operations at, for example, the conceptual vector or raster level, depends on such distinctions. We are currently involved in researching a full specification and classification of the conceptual vector-level operations.

Real-world Phenomena Model Level

Just as there is a tendency to mix conceptual vector-level concepts with the spatial object level, there is a tendency to mix conceptual raster-level concepts with those that involve real-world information. Thus, we introduce the levels involving such information first.

Often in research on automated generalization techniques, real-world information is either ignored (as in vector based methods), or viewed as "attributes" of a location (as in raster based methods). This may be adequate when spatial-object manipulations are the primary focus, but, increasingly, the meaning of the spatial object is being viewed as an important factor in generalization. Once real-world information, as well as spatial-object descriptions, are available to the system, more powerful generalization techniques may be developed.

To this end we have included a "Real-world Phenomena Model Level," and distinguish it clearly from the "Conceptual Spatial Object Level." The real-world phenomena model level includes any and all information that is deemed important enough to remember or utilize in generalization, but that is not part of the geometric or spatial description of real-world objects. Thus, the modeling of the river that a line stands for occurs at the real-world phenomena-model level, but the modeling of a line may occur at the conceptual spatial-object level. Or, a model of land use that is based upon classification of reflectance values belongs at the real-world phenomena model level, but the raster cell that holds specific attribute values does not. A fundamental class of activity that occurs at the real-world phenomena model level, and is important in generalization, is classification.

The Entity Object Level

This level is introduced to show the relationship between real-world phenomena modeled at the real-world phenomena model level, and spatial objects representing them, which are modeled at the conceptual spatial-object level. The name "entity object" is borrowed from the SDTS. In there, it was used to distinguish between a spatial object per se and a spatial object that is used to represent a real-world entity (or feature). Thus, in our figure, transformations that belong at the entity-object level are those that involve both the spatial object and whatever it represents. An open research issue concerns the identification of the major categories of transformation that may occur at this level. However, we can begin thinking about transformations at this level by analyzing the functions that are typical of the lower levels.

In contrast with many conceptual vector-level generalization operators, operators developed for use with the raster model actually perform functions at the entity-object level. That is, since they use and/or transform both real-world attribute data as well as structural spatial data, they occur at the level where the two types of data are bound. Although we describe below how some existing raster generalization operators actually exemplify activities at this level, the most commonly discussed vector generalization operators do not. However, there are some operations in commercial vector-based geographic information systems (GIS) that possibly should be classified at the entity-object level, in that spatial object descriptions are changed based on real-world types of attribute values. For example, although not typically viewed as generalization, the set operations in Arc/Info allow overlay and union of polygons, wherein attributes are used in specifying the operations, and new polygons are produced as the result.

The Conceptual Raster Level

Having described the levels that relate to real-world information, we can now identify the basic functional activities that occur at the raster level. Existing raster generalization operators help illustrate the functional activities. Three basic categories of functions that occur at the raster level are: change structure of cells, change attributes of cells, and change structure via attributes. The change in structure of cells may affect either cell size or cell number. The change in attributes of cells may involve data from neighboring cells or not. Due to potential attribute changes, the change to cell structure via attributes may also involve data from neighboring cells or not.

Following are examples of existing types of raster generalization operators, which are described in the McMaster and Shea framework, that fall into each of these functional categories:

CHANGE STRUCTURE OF CELLS
- change cell size
  resampling
- change total cell number
  simple structural reduction

CHANGE ATTRIBUTES OF CELLS
- use data from neighboring cells
  low pass filters and high pass filters
  edge enhancement
  feature dissolve, erode smooth, gap bridge
- use no data from neighboring cells
  categorical merge
  numerical categorization (image classification)

CHANGE STRUCTURE VIA ATTRIBUTES
- use data from neighboring cells
  categorical aggregation
- use no data from neighboring cells

Some of the above listed operators may also be classified as entity-object level operations. That is, any operations that require both attribute data and structural data must operate on the level where the two are bound. The clearest examples are methods that use data from neighboring cells. In contrast, methods that only change the structure of cells would not be classified at the entity-object level, as they perform a function strictly at the raster level.

At the purely raster level these functional groups of activity appear to be complete. However, raster data may be used to implement concepts at the spatial-object or entity-object levels. In order for this to occur, the notion of a "raster object" must be supported (Schylberg 1993). By this we mean that the system must be able to recognize a collection of contiguous raster cells, forming any shape, as an areal object. Raster object identification and management, typically not supported in GIS's, might be viewed as a raster level functional activity, but its purpose would primarily be to serve higher levels of information management.

Spatial Object Level

The "Conceptual Spatial Object Level" supports descriptions and transformations of objects in space that are based on the spatial object primitives, the point, the line, and the area. Since general spatial descriptions often also include spatial relationships and topology, this type of information belongs at this level, regardless of whether they are implicit or stored directly, possibly as attributes. Note that while attributes may occur at this level, the meaning of a spatial-object attribute has some spatial connotation. This does not imply that spatial objects never have real-world attributes associated with them. However, if an attribute with real-world meaning is directly associated with a spatial object, it means that an entity object supporting the connection between the two has effectively been created.

The transformations that support generalization at the spatial-object level include, in the sense that they subsume, the spatial/structural transformations of lower levels, such as vector and raster. However, at the spatial-object level, two lower level operators performing the same conceptual spatial operation would be redundant. Thus identifying, naming, and classifying operators at the conceptual spatial level should be done in a way that groups similar functions, and provides one name for like functions.

Although we are in the process of researching a full specification of functional operations for this level, this is not yet completed. However, we have identified some basic categories of function or activity. These categories are based upon generalization goals that are achieved through operators that manipulate spatial or geometric objects.

Specific algorithms have been developed that attempt to mimic traditional generalization techniques, e.g. simplification for weeding unnecessary information, smoothing for improving the aesthetic qualities, and displacement for shifting features apart. One major objective of generalization is thus for the aesthetic graphic display of cartographic information. Operations such as displacement, smoothing, and enhancement are designed for display purposes and can be classified under a display category of activity. A second objective is to reduce spatial and aspatial information. The reduction may be controlled either geometrically, as with simplify, collapse and structural reduction operations, or by making use of attribute data, as with low-pass and high-pass filters. The third objective of generalization, fusion, which joins or combines, may also be controlled either geometrically (as with aggregate, merge, and amalgamate operations), or by use of attributes, as with parallelepiped and minimum-distance-to-means operators. At a conceptual spatial level, then, basic classes of generalization functions include aesthetic display, reduction of information, and fusing of spatial features.

Notice that at the higher conceptual-spatial level, these basic classes are relevant for both vector and raster data. In addition, similar activities are grouped under a single classification. For instance, in dealing with objects in vector mode, a user amalgamates areas (which possibly represent census tracts), merges two linear features together, (which may represent the banks of a river), or aggregates point features into one area. All three, of course, fuse spatial objects together, whether they are points, lines, or areas. Likewise, a polygon dissolve in raster-mode fuses features together. Below, we depict this simplified structure and give examples of how existing operators may be classified.

DISPLAY
- *displace*
- *smooth*
- *enhance*

REDUCE
geometrically-controlled
- *simplify*
- *collapse*
- *structural reduction*

attribute-controlled
- *low-pass filters*
- *high-pass filters*

FUSE
geometrically-controlled
- *aggregate*
- *merge*

attribute-controlled
- *amalgamate*
- *minimum distance to means classification*
- *parallelepiped classification*
- *merge categories*

Looking at this categorization, it is obvious that, sometimes, the means of accomplishing a spatial-object transformation is through the use of attributes. That provides a clue that some of these operations may actually be working at the entity-object level. It also indicates that some of these operators have more than one function embedded in them. To put it another way, a specific kind of spatial-object transformation should be viewed as one function, and an attribute change or manipulation should be viewed as another, even if the two functions are accomplished with one operator at a lower level.

The conceptual spatial-object level should have transformations identified and defined for this level, that strictly deal with spatial-object changes. Once we have completed the identification and description of basic transformations for the spatial-object level, it will become easier to define transformations that are fundamental to the entity-object level.

This should be the basis for providing general support for integrating semantic information in with geometric descriptions of concepts.

CONCLUSION

In order to develop cartographic generalization operators that make greater use of semantic information, and to develop systems that can effectively support such operators, a better understanding of the relationship between real-world information and spatial-object descriptions for them is required. In addition, a better specification of what higher level operators ought to do with respect to various types of semantic information is needed.

To this end, we have begun a process of analysis. As an initial step we have identified the different conceptual levels of meaning that are currently, although usually implicitly, involved in the functions that generalization operators perform, or are envisioned to perform. The explicit separation and recognition of these levels allows us to understand the basic types of transformations and functional activities that occur at each. With this understanding, development of generalization operators can be appropriately focused on the level of interest to the developer. In addition, requirements for lower level operators needed to support higher level activities can be more easily identified.

REFERENCES

Armstrong, Marc P. 1991. "Knowledge Classification and Organization," in *Map Generalization: Making Rules for Knowledge Representation*. Barbara Buttenfield and Robert B. McMaster, (eds.). Longman, United Kingdom, pp. 103-118.

Beard, M. Kate. 1991. "Constraints on Rule Formation" in *Map Generalization: Making Rules for Knowledge Representation*. Barbara Buttenfield and Robert B. McMaster, (eds.). Longman, United Kingdom, pp. 121-135.

Brassel, K.E. 1985. "Strategies and Data Models for Computer-Aided Generalization" *International Yearbook of Cartography*, Vol. 25: pp. 11-29.

Buttenfield, Barbara P. and Robert B. McMaster. 1991. *Map Generalization: Making Rules for Knowledge Representation*. Longman, United Kingdom

Mackaness, William A. 1991. "Integration and Evaluation of Map Generalization" in *Map Generalization: Making Rules for Knowledge Representation*. Barbara Buttenfield and Robert B. McMaster, (eds.). Longman, United Kingdom, pp. 217-226.

McMaster, Robert B. 1991. "Conceptual Frameworks for Geographical Knowledge," in *Map Generalization: Making Rules for Knowledge Representation*. Barbara Buttenfield and Robert B. McMaster (eds.). Longman, United Kingdom, pp. 21-39.

McMaster, Robert B. and K. Stuart Shea. 1992. *Generalization in Digital Cartography*. Resource Publication of the Association of American Geographers.

National Institute of Standards and Technology. 1992. *Federal Information Processing Standard Publication 173 (Spatial Data Transfer Standard)*. U.S. Department of Commerce.

Peuquet, Donna J. 1990. "A Conceptual Framework and Comparison of Spatial Data Models," in *Introductory Readings in Geographic Information Systems*. Donna Peuquet and Duane Marble (eds.). Taylor & Francis, New York.

Peuquet, Donna J. 1988. "Representations of Geographic Space: Toward a Conceptual Synthesis." *Annals of the Association of American Cartographers*, 78(3). pp. 375-394.

Schylberg, Lars. 1993. *Computational Methods for Generalization of Cartographic Data in a Raster Environment*. Photogrammetric Reports No. 60, Royal Institute of Technology, Department of Geodesy and Photogrammetry, Stockholm, Sweden.

A HYBRID LINE THINNING APPROACH

Cixiang Zhan
Environmental Systems Research Institute
380 New York Street
Redlands, CA 92374, USA
Telephone (909) 793-2853 Fax (909) 793-5953
Email czhan@esri.com

ABSTRACT

The proposed hybrid thinning approach consists of preprocessing, distance-transform based skeleton extraction, sequential thinning and post-processing. The preprocessing smooths line edges fills holes. The Euclidean distance transform is then performed, and skeletons are extracted via established lookup tables to produce unbiased center lines. Sequential thinning, which deals with nearly-thinned lines better than other approaches, is then applied to thin skeletons to single-pixel width. The post-processing removes spurs, connects disconnected lines caused by skeleton extraction, and extends eroded line tips. Large data sets can be handled. Experiments on contour, parcel and land use data are presented.

INTRODUCTION

Line thinning is important to data compression, raster to vector conversion and pattern recognition. The general requirements of line thinning include the quality of results, speed and ability to handle large images with limited memory. The quality of results include the preservation of geometric and topological properties [Lam et al, 1992], and visual acceptance for problem domains. To preserve the geometric properties the thinned lines should be the median axes of the original line features, maintain the original line lengths, and be clean without additional spurs. To preserve the topological properties, the thinned lines must preserve the connection of the original lines without disconnection and additional loops. The visual acceptance is highly dependent on applications, and may include line smoothness and junction appearance.

Various thinning algorithms (Peuquet, 1984, Lam et al, 1992) have been developed to satisfy these requirements with some requirements being emphasized for a particular problem domain. Thinning algorithms can be divided into the iterative and the distance-transform based. with the iterative further divided into sequential and parallel classes. The speed of iterative approaches, which iteratively peel the contours of thick lines based on the local properties within a moving window, are generally dependent on line width, and its performance in geometric preservation depends on scan direction. The sequential algorithms, which do peeling based on the line patterns in the current iteration, are generally

faster on sequential machines and preserve connection better than the parallel algorithms, but their results are often biased away from the scan directions. Parallel algorithms, which peel contours based on patterns of the previous iteration, makes using parallel processors possible, and are less sensitive to scan direction than the sequential. But to maintain connection, they are forced to use sub-iterations or a large moving window. The distance transform based approaches, which normally perform Euclidean distance transformation on line network and extract skeletons based on the global information of distances from edges, may produce well centered thinned lines of width of one or two pixels at once. The resulted skeletons, however, may not preserve connection, and are sensitive to noise.

Hybrid approaches may be adopted to take advantages of different approaches. Arcelli and Sanniti (1985) combined distance transform and sequential thinning, with capability of reconstruction of original features.In this study, a hybrid thinning approach, uses lookup tables for skeleton extraction based on the Euclidean distance transform and perform sequential thinning and extensive post-processing to solve the problems inherent to the distance-transform based approaches, is described.

OVERVIEW OF THE HYBRID APPROACH,

In the hybrid approach, the Euclidean distance transform is performed first to produce x and y displacements of pixels from line edges. Skeletons are identified using lookup tables by checking the x and y displacements of pixels. Because the distance transform is sensitive to noise, a morphological dilation/erosion filter is optionally used before distance transform to smooth ragged edges and remove small holes within lines. Sequential thinning is followed to further thin skeletons to single-pixel width. The thinned lines are further processed by removing spurs, connecting broken skeletons, extending eroded line tips within the boundaries of the original line, and removing some false junction pixels. The program allows the control of the output line type being either smooth lines or lines with sharp corners. Large images are processed in strips with proper overlap between stirpes. The maximum thickness of input line features, as an input parameter, is used to determine the length limit of spurs and overlapping size of image strips. We will refer to line features in the context of a foreground consisting of black pixels, and a background consisting of white pixels.

DISTANCE TRANSFORM

In the distance transform, the white pixels in the background are used as the source, and the proximity of the black pixels on line features to source pixels are measured. The Euclidean distance transform calculates for each black pixel its X and Y displacements to its nearest source pixel. Actual distance calculation is avoided to reduce computation.

Danielson's algorithm [Danielsson, 1980] is used in the distance transformation. Two passes are required in distance mapping and five neighbors need to be visited in each pass. According to Danielsson, the errors from the true Euclidean distances with eight neighbors is sparsely distributed, and errors are bounded to be less than 0.076 pixel.

SKELETON EXTRACTION

We first briefly explain what we mean by the skeleton of a raster feature. In the raster domain, the disk of radius R centered at a pixel is the set of all pixels whose centers are within the distance R from the center of the pixel. Pixels that are exactly at distance R from the disk center are excluded from the disk. From the center of each feature pixel, there exists a disk of maximum radius among all disks which lie within the feature. The skeleton of a feature consists of those pixels whose maximum disk within the feature can not be covered by the maximum disk of another pixel in the feature.

For a disk of distance R, perform the Euclidean distance transform. Since the skeleton of the disk is its center pixel, it can be assumed the maximum disk of the neighbors of the center pixels is covered by that of the center pixel. If a pixel in the input image has X and Y displacements that match those of one of neighbors of the center pixel in the disk, we may assume the pixel is not a skeletal pixel in the input image. By checking the X and Y displacements of each black pixel and its neighbors it is possible to identify if the black pixel is a skeletal pixel of a line feature. Danielsson [1980] shows that the error by using this local neighborhood checking of X and Y displacements is extremely small due to raster points available.

To quickly identify skeletal pixels with this approach, a set of look-up tables with a pair of X and Y displacements as inputs, and the X and Y displacements of pixels, whose disk are covered by the disk in the diagonal or orthogonal direction, as outputs are established. To build those look-up tables, distance transform is performed on each of the disks with integer squares of radii from 1 to N, where $\sqrt{N}$ is the width of the thickest line the program is capable of processing. The quadrants of possible X and Y displacements of the disk of squares of radii 1 to 25 are shown in Fig. 1. The X and Y displacements of the disk centers become the input to the look-up tables and the X and Y displacements of its orthogonal or diagonal neighbors becomes the output of the orthogonal or diagonal look-up tables, respectively. Given the X and Y displacements of a pixel and its neighbors, the pixel can be identified as a non-skeletal pixel if its X and Y displacements are the outputs of either the orthogonal or diagonal look-up table with the X and Y displacements of one of its neighbors as the inputs.

To make the extracted skeletons adapted for line thinning, attention has been given to the following four situations in

building these look-up tables.

a. At a radius, the X and Y displacements may have multiple pairs of values, excluding the swap of X and Y displacements. For instance, at radius 5 (Fig. 1), both X and Y displacements (5,0) and (4,3) are valid, and may result from different scan directions. All multiple pairs of X and Y displacements at a radius should be identified.

b. A few pair of X and Y displacements at a disk center have different pairs of X and Y displacements at their diagonal neighbors. For instance X and Y displacements (6,0) at the disk center may have both pairs of X and Y displacements (5,0) and (4,3) at its diagonal neighbors depending on scan directions. For some pairs of input X and Y displacements is may be necessary to have two pairs of X and Y displacements as their outputs.

c. Border pixels (1,0 or 0,1) need special handling in order to maintain connection of thin lines. As shown in Fig. 2 the border pixel of (1,0), marked by a thick box, is not normally a skeletal pixel, but should not be removed in order to maintain necessary connection. Those border pixels, which have no neighbor pixel with X and Y displacements (1,1) or (2,0) on one side and another neighbor pixel with X and Y displacements (0,0) on the opposite side, should be removed.

d. To further reduce the disconnection caused by skeleton extraction, the disk of X and Y displacements (2,0) is made not being covered by that of X and Y displacements (3,0). This pixel of (2,0), marked with a thick box in Fig. 2, together with the pixel of X and Y displacements (1,0), also marked with a thick box, maintain the connection in a line with a narrow portion (Fig. 2). Further modifications in look-up tables may improve connection, but tends to increase noise and blur the major line features in skeletons.

It is well known that the skeleton of a line feature extracted using the median axis transform in the discrete space may be of two-pixels width and disconnected, and may have spurs at turns and line ends. For the purpose of line thinning, the erosion of line ends during skeleton extraction need to be recovered. The following four sections describe the processing steps to solve these problems.

SEQUENTIAL THINNING

The major purpose of the step is to further thin the resulted skeletons into thin lines of one-pixel width. In addition single black pixels resulted from skeleton extraction are dropped, and gaps of one-pixel width between disconnected skeletons are filled. The basic idea of the sequential thinning algorithm by Greenlee [1987] is used since the algorithm takes into account of complete 256 junction patterns and is flexible in adjusting the rules of pixel elimination. First the resulted skeleton image is encoded.

Each neighbor of a pixel are assigned an direction number of N^2 where (1 <= N <= 8) depending on the direction of the neighbor. The code of a pixel is the sum of the direction numbers of its black neighbors. Black pixels of code 0 is single pixel and are dropped during encoding. Decision rules for pixel filling and peeling are developed based on the coding. When a pixel is filled or removed, the codes of its neighbors are updated. The procedure is iterated until no change occurs. Peeling rules are set differently for smooth and sharp-turned output lines. Since the skeletons are of no more than two pixel width, the bias of thinned lines that may be caused by sequential thinning are negligible. One pass is enough to thin all skeletons to one-pixel width. At least four passes are required in the step, one for encoding, one for filling, one for thinning and one for checking code changes. Approaches based on contour processing [Kwok, 1988], [Naccache, 1984] may be used for those nearly thinned lines to improve efficiency.

SPUR REMOVAL

Before spur removal, all black pixels are encoded using the number of its black neighbors, and white pixels are labeled 0. A Pixel of code 1 is a tip of a line, a pixel of code 2 is generally the intermediate line pixel, and a pixel of code 3 or larger generally is a junction pixel. Sometimes, a pixel of code 2 may be a spur pixel (Fig. 3(d), and a pixel of code 3 may forms a false junction, as shown in Fig. 3(e).

There are two types of spurs often occur in the resulted skeletons, 1) middle spurs that occur in the middle of a line when the line direction or thickness changes and 2) end spurs that occur at the line ends (Fig. 3) Middle spurs can be processed individually. End spurs, however, appear as pairs, and each pair must be processed simultaneously. Otherwise, after the first spur is removed, the second spur becomes a portion of a line, and can no longer be recognized. The spurs may also be classified as code 1 spurs whose tip pixels are coded 1, and code 2 spurs whose tip pixels are coded 2 and whose length is one pixel. We will discuss the removal of code 2 spurs in the next section.

To identify code 1 spurs, the resulted output is scanned to find pixels of code 1 (tips). From a line tip the next black pixel along the line is searched. When a pixel of code 2 is found, its another black neighbor is searched. The search is restricted in a limited sector without visiting the neighbors of the preceding black pixel. Searching ends when reaching a junction pixel (code > 3), or an end pixel (code 1 or 0), or the defined maximum spur length are reached. The maximum spur length can be defined as 0.8 times the maximum line thickness. When a search ends at a pixel of code 3 or larger, we call the pixel the junction-end pixel of the search or the spur that is identified. The major difficulty is to identify false junctions in Fig. 3(e).

Removing spurs that does not end at a junction-end pixel is to simply to remove all pixels visited during search. When a spur ends at a junction-end pixel and spur is identified, however, one must to decide if the junction-end pixel of the spur should be removed. When the junction-end pixel constitutes a necessary connection for the remaining line, it should not be removed. On the other hand, leaving an extra pixel at a junction means that the spur is not completely removed.

To identify false junctions and determine the removal of junction-end pixels of spurs, all possible patterns when search reaches a pixel of code 3 or larger (a true or false junction) are studied.

Fig. 4 shows the all possible junctions patterns when reaching a junction pixel of code 3 from the upper-left and from the top. For reaching junctions of codes larger than 3, the pattern can be obtained similarly. Based on the analysis of those patterns the following rules are developed to identify spurs and to handle the junction-end pixels. In pattern 1, 2, 3, 5 and 7 of Fig. 4 true junctions are reached, and spurs are identified. The junction-end pixel J should not be deleted, because the other two neighbors of pixel J, (except for the neighbor preceding pixel J in the search) are not contiguous, i.e. at least a white neighbor is between those neighbors.

In pattern 4, 6 and 8, the other two neighbors of pixel J are contiguous, and among them one and only one is the direct (orthogonal) neighbor of pixel J (labeled D in Fig. 4). The following rules are developed to detect false junctions. If the code of D is less than 3, a false junction is detected; if the code of D is larger than 3, a true junction is identified. When the code of D is 3, J and D form a false junction if the rest of neighbors of D are not contiguous, and J reaches a true junction otherwise. Whenever a true junction is reached, the junction-end pixel J can be removed.

When the code of a junction-end pixel is larger than 3, a true junction is always reached, and a spur is identified. To determine if the junction-end pixel is to be removed, the rules for junction end pixels of code 3 can be similarly applied. If the rest of neighbors of a junction-end pixel are not all contiguous, the junction-end pixel is not removed, otherwise, it can be removed.

Although some junction-end pixels, such as in pattern 5 of Fig. 4, can be removed without creating disconnection, it is better to maintain them for the simplicity of rules, and for better junction handling by considering the rest of junction pixels and different junction handling requirements.

When a spur is removed, the codes of neighbors of the last pixel of the spur should be updated. when the code of one of its neighbor becomes 1, a new line tip is created and the need

to be extended

REMOVAL OF EXTRA PIXELS OF CODE 2

After the previous processing some pixels of code 2, which are actually spurs of one-pixel length or the corners of sharp turns that are redundant for smooth line requirement, need to be removed. There are two types of pixels of code 2 that need to be removed, as shown in Fig. 3(d) and (e). The one in Fig. 4(e) has two adjacent direct neighbors, and is not subject to removal when sharp turns are demanded. The spur in Fig. 4(d) can be easily identified since this type of pixels of code 2 always has two adjacent neighbors. The code 2 spurs may also appear as pairs at line ends (Fig. 3(c)), and each pair needs to be handled together. In all other cases pixels of code 2 form necessary line connections, and should not be removed. When a pixel of code 2 is removed, the codes of its neighbors should be updated. When the updated code of a neighboring pixel becomes 2, the pixel needs to be examined for removal, and the process becomes recursive. When the updated code becomes 1, the new line tip is to be extended, as shown in the next section.

EXTENDING LINE TIPS

The purpose of this step is to extend the tips of the thinned lines, to the boundaries of the original lines, or to connect to other lines. In the first case, we try to recover the line length eroded in the previous processing, and in the second case, we try to link the disconnected skeletons. The extending directions are determined in increments of 22.5 degrees by using the last 3 pixels from the line tips. Extending in the directions of angles 22.5+45*N (n=0,7) is realized by alternatively extending in the orthogonal and diagonal directions. Sixteen extending directions makes extended lines coincide with the original line direction better than eight extending directions do. The process that extends line ends are embedded in the process of spur removal and is invoked when the updated code of any pixel becomes 1.

HANDLING LARGE IMAGES

To be efficient many processing steps require the input and output images to be entirely held in memory. When a large image can not be held in memory, it can be processed strip by strip. Each strip spans the width of the image, and overlaps its adjacent strips by a number of rows, which is no less than the maximum thickness of line features. No significant problems have been found in the output by processing in strips. Only a shift of one pixel may be found when some vertical lines cross the border between strips.

CONCLUSION

The paper proposes a hybrid thinning approach, in which the distance-transform based approach is combined with the

sequential approach. The hybrid approach takes the advantages of both approaches: the speed independent of line width and fine median axes from the Euclidean distance transform, and flexibility to handle the fine detail of nearly thinned lines from the sequential approach. The look-up tables for skeleton extraction enable skeletons to be quickly and properly extracted for further processing. The lookup tables have been established to be able to handle thick lines of width up to 60 pixels. To solve the problems left mainly by the skeleton extraction based on distance transform, such as spurs and disconnection, extensive post-processing procedures are developed. Simultaneously handling of pairs of spurs, and recursively processing new spurs are essential to ensure the quality. Extending line tips in proper directions not only restores the line length, more importantly, re-links the disconnected line skeletons.

The approach performed well on contour lines, parcel map, roads and various land-use and land-cover data, where the range of line widths may not be regular, features may be relatively noisy, and sharp-turned or smooth lines may be required. Some of the results are show in Fig. 5. The major problems with the approach exist at junctions of thick lines, where it produces dimples at T junctions, and creates two junctions at a X junction. These junction problems can be better handled in vector structures if vertorization is performed after thinning.

REFERENCES

Arcelli, Carlo and Gabriella, Sanniti DI Baja, 1985, A Width-Independent Fast Thinning Algorithm, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.7:463-474.

Danielsson, Per-Erik, 1980, Euclidean Distance Mapping, *Computer Graphics and Image Processing*, Vol. 14:277-248.

Greenlee, David D, 1987, Raster and Vector Processing For Scanned Line Wrok, *Photogrammetric Engineering and Remote Sensing*, Vol. 53:1383-1387.

Kwok, Paul C.K., 1988, A Thinning Algorithm By Contour Generation, *Communication of the ACM*, Vol. 31 1314-1324.

Lam, Louisa, Seong-Whan Lee, and Ching Y. Sun, 1992, Thinning Methodologies - A Comprehensive Survey, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, Vol. 14:869-885.

Naccache, N.J. and Shinghal, R, SPTA: A proposed algorithm for the Thinning Binary Patterns, *IEEE Trans. on System, Max and Cybernetics*, SMC-14:409-418.

Peuquet, Donna J., 1981, An Examination of Techniques for Reformatting Digital Cartographic Data /Part 1: The Rater-toVector Process, *Cartographica*, Vol. 18:34-48.

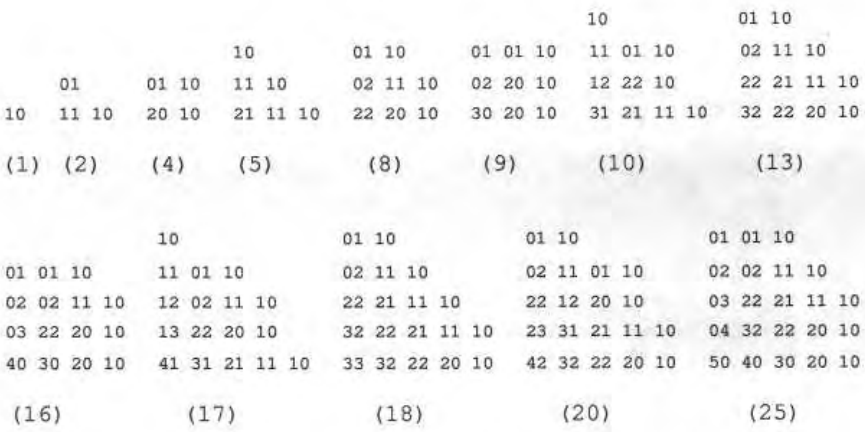


Fig. 1 X and Y Displacements of Disks (The numbers in parentheses are the squares of radii of disks)

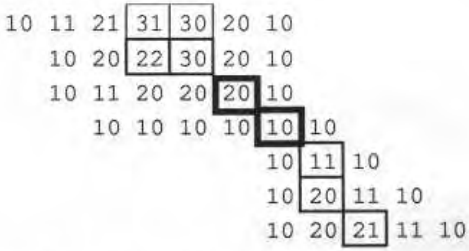


Fig. 2 Modifications to Normal Skeleton Extraction. Normal skeleton pixels are marked with thin boxes, Pixels with thick boxes are put into skeletons.

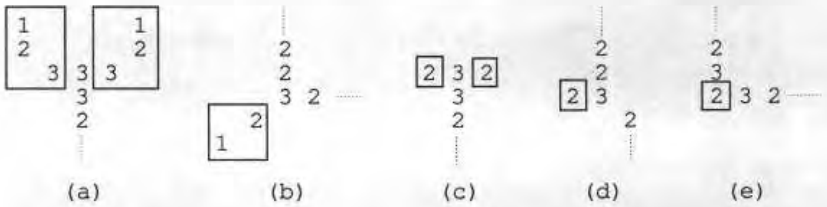


Fig. 3 Types of Spurs. (a) and (b): code 1; (c), (d) and (e) code: 2. (a) and (b): end spurs; (b) and (d): middle spurs. (e) middle spur in smooth lines. Pixels surrounded by boxes are spur pixels.

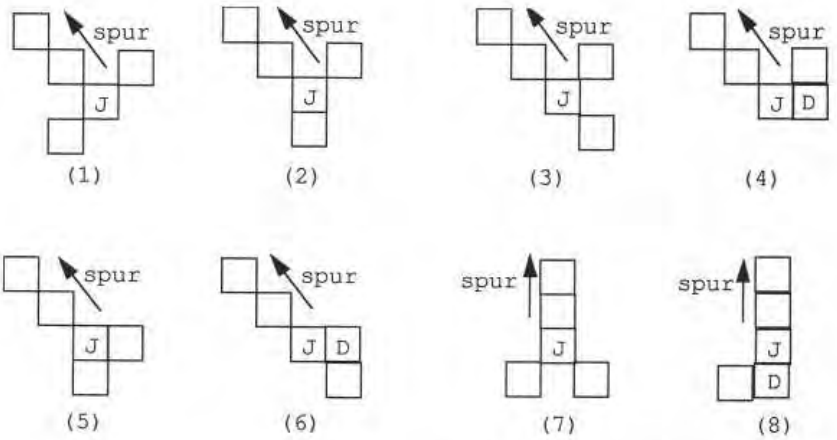
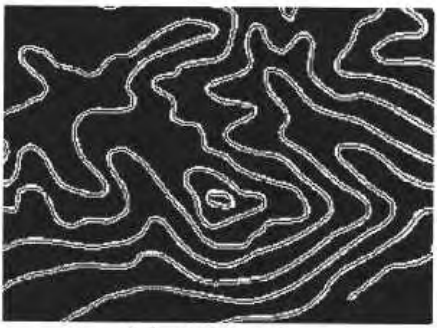
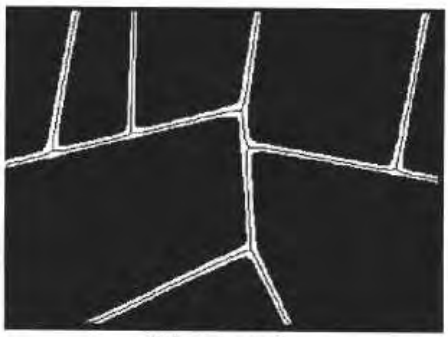


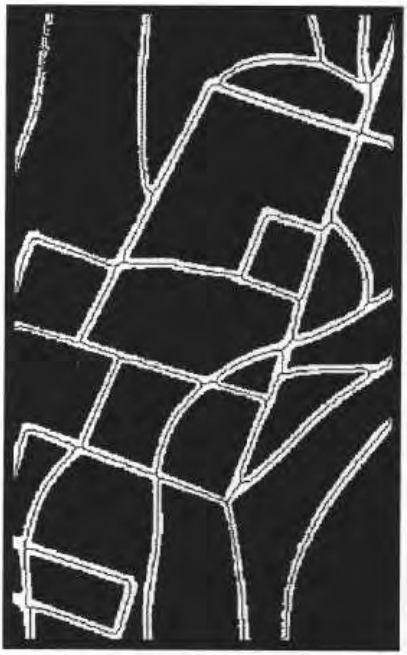
Fig. 4 Patterns When Search from Tips Reaches Pixels of Code 3 (Labeled J) from the Upper-Left (1)-(6), and the Top (7)-(8). D is a direct neighbor of J.



(a) contour



(b) parcel



(c) land use

Fig. 5 Thinning Results of Line Features

Conflict Resolution in Map Generalization: A Cognitive Study

Feibing Zhan and David M. Mark
National Center for Geographic Information and Analysis
Department of Geography
State University of New York at Buffalo
Buffalo, NY 14261
zhan@geog.buffalo.edu, geodmm@cc.buffalo.edu

ABSTRACT

The generalization of a single type of feature such as linear or areal feature has been addressed in digital cartography. In real world applications, generalization almost always involves more than one type of feature. Assuming different types of features are generalized separately, an important issue is how to resolve the conflicts among different types of generalized features when they are assembled back together. The cognitive process concerned with this type of conflict resolution is not well understood. This is one of the major obstacles for developing a knowledge-based system for map generalization. This paper attempts to explore the process of conflict resolution using a human subjects experiment, focusing on the resolution of conflicts between linear and areal features. The experiment is designed as follows. First, a number of areal features published in the literature and a linear feature are obtained. Second, the two types of features are generalized to a certain scale separately, and then are assembled back together. Third, the results before and after generalization are given to different subject independently, and each subject is asked to identify the conflicts between the linear and areal features, and to rank several suggested strategies for resolving the conflicts. Preliminary results suggest that the violation of topological relations is the major source of conflicts, and the best strategy to resolve the conflicts is to displace the generalized line locally.

INTRODUCTION

There have been extensive research efforts on the generalization of a single type of feature such as linear or areal feature, and significant results have been achieved (MacMaster, 1986, 1987, 1989; Muller, 1990; Buttenfield and McMaster, 1991; Muller and Wang, 1992; Zhan and Buttenfield, 1993). In recent years, attention has been paid to the development of more comprehensive map generalization system in order to facilitate real world applications which involve the generalization of various types of map features, namely, point, linear and areal features (for example, Brassel and Weibel, 1988; McMaster and Shea, 1988; Shea and MacMaster, 1989). Although some theoretical and conceptual models have been proposed for such a system (Mark, 1989, 1991; Armstrong and Bennet, 1990; Armstrong, 1991; Shea, 1991), no comprehensive operational map generalization system has been realized. One of the major difficulties for developing such a system is the lack of full understanding of map generalization processes involving more than one type of feature. What is, for example, the process for generalizing a map containing three types of features? How does an expert cartographer identify the conflicts among different types of generalized features? How are the conflicts resolved by a cartographic specialist? These are

questions that should be addressed before a comprehensive map generalization can be fully realized.

This paper explores the human cognitive process for conflict resolution in map generalization through human subjects experiments. Here conflict is defined as the violation of spatial relations during map generalization. When different types of features are generalized separately, an important issue is how to resolve the conflicts among different types of features when they are assembled back together. The cognitive process concerned with this type of conflict resolution is not well understood, and remains as a major obstacle for fully automate the process of map generalization. We will focus on the process of identifying conflicts as well as the process for conflict resolution between linear and areal features in this discussion. We will first give a brief discussion of spatial relations and its connection with conflict resolution in map generalization.

SPATIAL RELATIONS AND CONFLICT RESOLUTION

The basic idea behind map generalization in most cases is easy to state: reduce the complexity of visual information in such a way as to retain important aspects and suppress unimportant ones. Some of the things to preserve are geometric, and McMaster (1987) has done an excellent job of evaluating various line generalization procedures, primarily according to geometric summary measures. But cartographers have also always be concerned with preserving the semantics of maps, especially spatial relations. To formalize this, it is necessary to provide formal definitions of spatial relations as well.

Recently, Max Egenhofer and his colleagues have developed new ways to characterize topological spatial relations. This approach is termed the '9 intersection' because it identifies an interior, a boundary, and an exterior for each entity, and then tests which of the 9 possible intersections (3 parts times 3 parts) are non-empty. The 9-intersection model was first proposed by Egenhofer and Herring (1991), and has been shown to have much more power to characterize spatial relations than had previous formal models of spatial relations (see Egenhofer et al., 1993). It has also been found to be closely related to the ways that people characterize and distinguish spatial relationships in language (Mark and Egenhofer, 1992; Mark and Egenhofer, under review).

The major concern of the present paper is how to preserve spatial relations between a line and some regions as both are generalized. For relations between a line (simple, unbranched) and a region (simple, connected region with no holes), Egenhofer and Herring (1991) found that 19 topologically-distinct spatial relations between such entities could be distinguished, depending on where the ends and the body of the line lie in relation to the parts of the region. Ideally, the topological spatial relation, as defined by the 9-intersection, should be the same after generalization as it was before. And if this is not possible, the spatial relation after generalization should be a 'conceptual neighbor' of the relation before.

For long lines that pass through the study area, such as those examined in this study, only those spatial relations with both ends of the line outside the region remain relevant. Of the 19 relations distinguished by the 9-intersection model, only 3 would meet this criterion, and these are differentiated by whether the line is entirely disjoint from the region; or the body of the line intersects with the interior of the region; or the body of the line just touches or is co-linear with the region's boundary. The performance of an algorithm, or the judgements of a cartographer, can be based in part upon whether the spatial relation changes among these categories from before to after the generalization operation(s). If subjects tend to preserve spatial relationships, then an algorithm or rule which does not should be modified, or a post-generalization displacement or geometric adjustment should be performed to re-establish the 'correct'

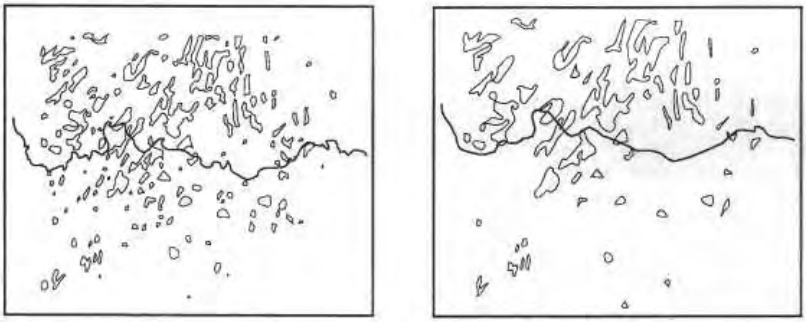


Figure 1. An example of the stimuli used for conflict identification

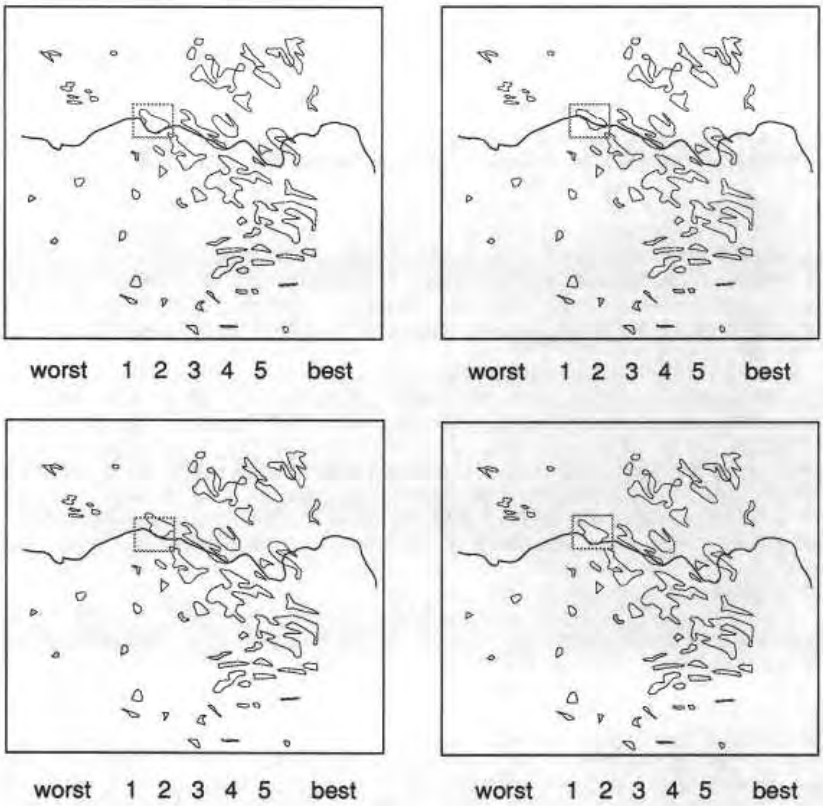


Figure 2. An example of the stimuli used for testing the strategies for conflict resolution. A subject is aksed to rank the results from the four strategies.

spatial relation. On the other hand, if some changes in topology bother the human subjects whereas others do not, rules could be developed to 'fix' only those cases which contradict the human tendencies. Moreover, some cartographers suggest that local displacement of the generalized line is probably the most common strategy for resolving conflicts between linear and area features. But more rigorous evidence is needed in order to support this intuition. We will now turn to discuss the experimental design.

EXPERIMENTAL DESIGN

In order to obtain the data for the experiment, a number of areal features published in the literature and a linear feature from real world data were obtained. Then, the linear feature was overlaid with the areal features in several different ways in order to obtain a number of possible spatial relations between the linear feature and the areal features before they are generalized. The two types of features were further generalized to a certain scale separately, and then assembled back together. The areal features and their generalized versions are from Muller and Wang (1992). The linear feature is from Zhan and Buttenfield (1993). The subjects used in the experiment were graduate students at the Department of Geography, State University of New York at Buffalo. All of them had some training in cartography and/or GIS.

The experiment consisted of two parts. Part I was intended to test the process of identifying conflicts between linear and areal features. Part II was used to test what strategies a person is most likely use to resolve the conflicts.

The following written general instructions were used at the beginning of the experiment in order to give the subjects some basic background information about the experiment.

> *"There are two versions of map features used in the experiment: the original one and the generalized one. Each of them has a linear feature and a number of polygonal features. The linear feature and the polygonal features are generalized first, and then assembled back together. It is assumed that the polygonal features and linear features are 'correctly' generalized, and our goal here is only to resolve the conflicts between the linear features and the polygonal features."*

The following written instructions were given to the subjects in Part I of the experiment, followed by two diagrams showing features before and after generalization. Figure 1 is one example of the diagrams.

> *"Each figure on the following pages shows a version of original features and a version of generalized features. Please identify (circle) the conflicts (anything that you do not feel is right)* ***between the linear features and the polygonal features*** *in the generalized features (lower portion of the diagram)."*

At the end of Part I, a subject was asked to "(Please) describe in writing how you found the conflicts."

In Part II of the experiment, the following written instructions were given to the subjects first.

> *"Each figure on the following pages shows a number of suggested strategies for resolving the conflicts* ***between the linear features and the polygonal features*** *in the generalized features. Please rank (choose a number between 1 and 5) the suggested strategies for the conflict resolution. Please use the relevant version of original*

features as reference if necessary."

Then two groups of diagrams are presented to a subject. Figure 2 is one example of the diagrams in the first group. The spatial relations between the linear and areal features are the same, but conflict to be resolved in each diagram is different. The first group consists of 3 diagrams, and the second is composed of 2 diagrams. The original features (before generalization) for each group was provided to a subject for reference purpose. In each diagram, results of conflict resolution by four suggested strategies were presented to a subject. These four strategies are: (a) modifying the geometry of the linear feature locally, (b) modifying the geometry of the areal feature locally, (c) displacing the linear feature locally, (d) modifying the geometry of the linear feature and the geometry of the areal feature locally. The subject was asked to rank the result of each of the suggested strategies. At the end of Part II, each subject was asked to "(Please) describe in writing how you would resolved the conflicts, if you think you could do better than any of the examples (use a drawing if necessary)." The results of the experiment are discussed in the following section. So far we have obtained results from seven subjects.

RESULTS

Number of conflicts identified

The number of conflicts identified by the subjects varies widely. For the first diagram, the number varies from 1 to 5 with the mean of 3. For the second diagram as shown in Figure 1, the mean is 3.4, the highest number of conflicts identified is 6, and one subject identified no conflicts. This implies that conflict identification is very subjective.

Another interesting question is what type of violation of spatial relations is considered a conflict. There are general agreements among the conflicts identified by the subjects. In the first diagram of Part I, six subjects agree on one conflict, and another two conflicts were also identified by four subjects. In the second diagram as shown in Figure 1, two conflicts were identified by five subjects, and one conflict is identified by four subjects. The results of conflict identification are summarized in Figure 3 for the diagram shown in Figure 1.

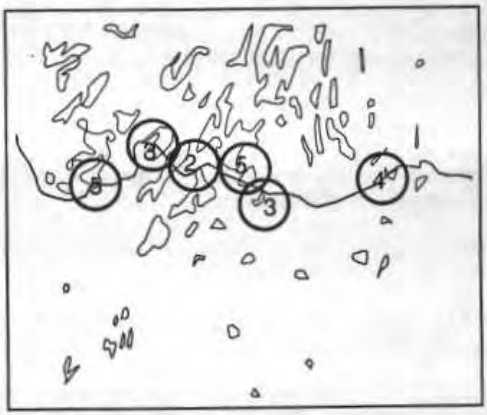


Figure 3. Number of subjects identifying it as conflict.

The process of identifying conflicts

It is clear from the subjects' written comments that the general process is to follow the linear feature and compare the spatial relations between the linear feature and the areal features in the versions before and after generalization. This suggests that relevant algorithm should try to mimic this process. All subjects wrote down the reasons why a conflict is identified. The reasons are all related to the violation of spatial relations between the linear features and the areal features after generalization as indicated in Section 2. In addition to the spatial relations, some subjects suggest that finer distinctions should be made, such as the distance between the linear feature and the areal features in the original maps must also be preserved proportionally in the generalized ones.

The best strategy

Five conflicts are used in Part II of the experiment. The results of conflict resolution of the five conflicts by four different strategies are evaluated by the subjects. The average score of the results by the four strategies is depicted in Figure 4. It can be seen from Figure 4 that the results by the strategy of "displacing the line locally" receives the highest score, 4.7. Thus it can be considered as the most recommended strategy. This is consistent with the general linguistic principle that linear features are normally located relative to areal features (rather than the other way around), and that the areal features are less movable. The results by the strategy of "modifying the areal feature locally" receives the lowest average score, 2.9, which may not be used at all. These results are in conformance with the general intuition. However, the other two strategies can not be excluded as evidenced in this experiment. The strategy of "modifying the geometry of the linear feature locally" is scored 4.0, and the strategy of "modifying the geometry of the linear feature and the geometry of the areal feature locally" got a score of 3.9. But the issue of which strategy to be used under what condition remains to be investigated.

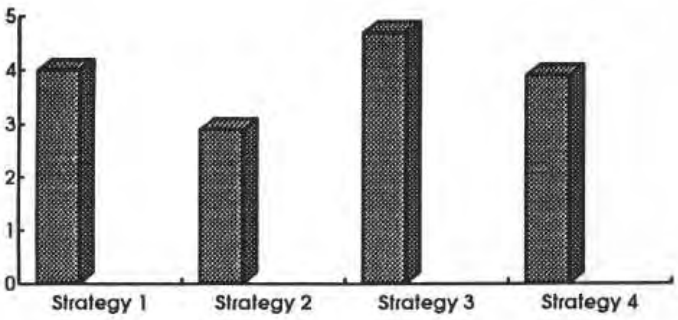


Figure 4. Mean scores of different conflict resolution strategies. Strategy 1: modifying the geometry of the linear feature locally, Strategy 2: modifying the geometry of the areal feature locally, Strategy 3: displacing the linear feature locally, Strategy 4: modifying the geometry of the linear feature and the geometry of the areal feature locally.

SUMMARY AND FUTURE WORK

Preliminary results clearly suggest that the conflicts arising in map generalization (violation of topological relations in this study) should be resolved, and among the four suggested strategies for resolving the conflicts, the results of displacing the generalized line locally received the highest rank. Although more experiments are needed in order to draw more rigorous conclusions on this, existing algorithms or rules should be modified in order to facilitate conflict resolution when more than one type of feature is involved during generalization.

Clearly, the experiment should be run using subjects with more cartographic generalization expertise, and should be repeated with larger samples. As the linear features and the areal features are taken out of context, the semantics of these features are not considered in this paper. The influence of the semantics on the identification of conflicts and on the strategies for conflict resolution is worth investigating.

For an automatic procedure for conflict resolution to be effective, it must accommodate the following issues: the identification of the conflicts, and the resolution of the conflicts. Either of these issues alone could be a challenging task, for instance, areal features before or after generalization can be completely different in the digital environment; to identify the conflicts, the procedure must be able to 'recognize' the corresponding objects in the original and generalized versions of features.

ACKNOWLEDGMENTS

This paper is a part of Research Initiative 8, "Formalizing Cartographic Knowledge," of the U.S. National Center for Geographic Information and Analysis (NCGIA), supported by a grant from the National Science Foundation (SES-88-10917); support by NSF is gratefully acknowledged. We would like to thank Dr. Zeshen Wang for supplying the digital generalized area-patch data.

REFERENCES

Armstrong, M. P., 1991. Knowledge classification and organization. Chapter 2.2 In Buttenfield, B. P. and McMaster, R. B., 1991. *Map generalization: making rules for knowledge representation.* London Longman.

Armstrong, M. P. and Bennett, D. A., 1990. A knowledge based object-oriented approach to cartographic generalization. *Proceedings GIS/LIS '90,* Anaheim California: 48-57

Brassel, K. E. and Weibel, R., 1988. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems,* 2 (3): 229-244

Egenhofer, M., and Herring, J., 1991. *Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases.* Technical Report, Department of Surveying Engineering, University of Maine, Orono, ME.

Egenhofer, M., Sharma, J., and Mark, D. M., 1993. A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis. *Proceedings, Auto Carto 11,* in press.

Mark, D.M., 1979. Phenomenon-based data structuring and digital terrain modeling. *Geo-Processing* 1: 27-36

Mark, D.M., 1989. Conceptual basis for geographic line generalization. *Proceedings Auto-Carto 9, Ninth International Symposium on Computer-Assisted Cartography,* Baltimore, Maryland, March 1989: 68-77

Mark, D.M., 1991. Object modeling and phenomenon-based generalization. Chapter 2.3 In Buttenfield, B. P. and McMaster, R. B., (eds.). *Map Generalization: Making Rules for Knowledge Representation.* London Longman.

Mark, D. M., and Egenhofer, M., 1992. An Evaluation of the 9-Intersection for Region-Line Relations. *GIS/LIS '92*, San Jose, CA, pp. 513-521.

Mark, D. M., and Egenhofer, M. J., 1993. Modeling Spatial Relations Between Lines and Regions: Combining Formal Mathematical Models and Human Subjects Testing. Under review.

McMaster, R.B., 1986. A statistical analysis of mathematical measures for linear simplification. *The American Cartographer* 13 (2): 103-116

McMaster, R.B., 1987. Automated line generalization. *Cartographica* 24 (2): 74-111

McMaster, R.B., 1989. The integration of simplification and smoothing algorithms in line generalization. *Cartographica* 26 (1): 101–121

McMaster, R. B and Shea, K. S., 1988. Cartographic generalization in a digital environment: a framework for implementation in a geographic information system. *Proceedings GIS/LIS '88*, San Antonio Texas, 1: 240-249

McMaster, R. B., 1991. Conceptual frameworks for geographical knowledge. Chapter 1.2 In Buttenfield, B. P. and McMaster R B., 1991. *Map Generalization: Making Rules for Knowledge Representation.* London Longman.

Muller, J. C., 1990. The Removal of Spatial Conflicts in Line Generalization. *Cartography and Geographic Information Systems* 17(2): 141-149.

Muller, J.C. and Z.-S. Wang. 1992. Area-Patch Generalization: a Comparative Approach. *The Cartographic Journal.* Vol. 29, pp. 137-144.

Shea, K. S., and McMaster, R. B., 1989. Cartographic generalization in a digital environment: When and how to generalize. *Proceedings Auto-Carto 9, Ninth International Symposium on Computer-Assisted Cartography*, Baltimore, Maryland, March 1989: 56-67

Shea, K. S., 1991. Design considerations for a rule-based system. Chapter 1.1 In Buttenfield, B. P. and McMaster, R. B., 1991. *Map Generalization: Making Rules for Knowledge Representation.* London Longman.

Zhan, F. and B. P. Buttenfield, 1993. *Multi-Scale Representation of a Digital Line.* Revised manuscript under review.

PARALLEL SPATIAL INTERPOLATION

Marc P. Armstrong
Departments of Geography and Computer Science and Program in Applied Mathematical and Computational Sciences

Richard Marciano
Gerard Weeg Computer Center and Department of Computer Science

316 Jessup Hall
The University of Iowa
Iowa City, IA 52242
marc-armstrong@uiowa.edu

ABSTRACT

Interpolation is a computationally intensive activity that may require hours of execution time to produce results when large problems are considered. In this paper a strategy is developed to reduce computation times through the use of parallel processing. A serial algorithm that performs two dimensional inverse-distance weighted interpolation was decomposed into a form suitable for processing in a MIMD parallel processing environment. The results of a series of computational experiments show a substantial reduction in total processing time and speedups that are close to linear as additional processors are used. The general approach described in this paper can be applied to improve the performance of other types of computationally intensive interpolation problems.

INTRODUCTION

The computation of a two-dimensional gridded surface from a set of dispersed data points with known values is a fundamental operation in automated cartography. Though many methods have been developed to accomplish this task (e.g. Lam, 1983; Burrough, 1986) inverse distance weighted interpolation is widely used and is available in many commercial GIS software environments. For large problems, however, inverse distance weighted interpolation can require substantial amounts of computation. MacDougall (1984), for example, demonstrated that computation times increased dramatically as the number of data points used to interpolate a small 24 by 80 map grid increased when a Basic language implementation was used. While little more than a half hour was required to interpolate the grid using 3 data points, almost **13 hours** were required when calculations were based on 100 points (see Table 1).

Table 1. Computation time (hours) for interpolating a 24 x 80 grid using an 8 bit, 2 MHz microcomputer.

N Points	Hours
3	0.57
10	1.51
25	3.50
100	12.46

Source: MacDougall, 1984.

MacDougall was clearly using an unsophisticated algorithm ("brute force") implemented in an interpreted language (Basic) which ran on a slow

microcomputer; most workstations and mainframes would now compute this problem in a few seconds. Using MacDougall's 100 point problem as an example, what took over 12 hours, can now be computed in just under 12 seconds (Table 2) using a single processor on a "mainframe-class" Encore Multimax computer.

Table 2. Computation time (seconds) for interpolating a 24 x 80 grid using 1 Encore processor.

Points	Seconds
10	1.50
100	11.92

The need for high performance computing, however, can be established by increasing the problem size to a larger grid (240x800) with an increased number of data points (10,000); this larger and more realistic problem roughly maintains the same ratio of points to grid points used by MacDougall. When the problem is enlarged in this way, a similar computational "wall" is encountered: using 3 proximity points, the interpolation problem required 7 hours and 27 minutes execution time on a fast workstation, an RS/6000-550. Based on the history of computing, this is a general pattern: As machine speeds increase, so does the size of the problems we would like to solve, and consequently there is a continuing need to reduce computation times (see e.g. Freund and Siegel, 1993).

Several researchers have attempted to improve the performance of interpolation algorithms. White (1984) comments on MacDougall's approach and demonstrates the performance advantage of integer (as opposed to floating point) distance calculations. Another important strategy reduces the total number of required computations by exploiting the spatial structure inherent in the control points. Hodgson (1989) concisely describes the interpolation performance problem and provides a solution that yields a substantial reduction in computation time. His method is based on the observation that many approaches to interpolation restrict calculations to the neighborhood around the location for which an interpolated value is required. Traditionally, this neighborhood has been established by computing distances between each grid point and all control points and then ordering these distances to find the *k*-nearest. Hodgson reduced the computational overhead incurred during these steps by implementing a new, efficient method of finding the *k*-nearest neighbors of each point in a point set; these neighbors are then used by the interpolation algorithm. Clarke (1990) illuminates the problem further and provides C code to implement a solution.

Despite these improvements, substantial amounts of computation time are still required for extremely large problems. The purpose of this paper is to demonstrate how parallel processing can be used to improve the computational performance of an inverse-distance weighted interpolation algorithm when it is applied to the large (10,000 point) problem described earlier. Parallel algorithms are often developed specifically to overcome the computational intractabilities that are associated with large problems. Such problems are destined to become commonplace given the increasing diversity, size, and levels of disaggregation of digital spatial databases.

The parallel algorithm described here is based on an existing serial algorithm. Specifically, we demonstrate how a serial Fortran implementation of code that performs two dimensional interpolation (MacDougall, 1984) is translated, using parallel programming extensions to Fortran 77 (Brawer, 1989), into a version that runs on a parallel computer. In translating a serial program into a form suitable for parallel processing, several factors must be considered including characteristics of the problem and the architecture of the computer to be used. We first consider architectural factors and then turn to a specific discussion of our parallel implementation of the interpolation algorithm; this "brute force" implementation represents a worst case scenario against which other approaches to algorithm enhancement can be compared. The approach described here can be applied to enable the use of high performance parallel computing in a range of related geo-processing and automated cartography applications.

ARCHITECTURAL CONSIDERATIONS

During the past several years, many computer architects and manufacturers have turned from pipelined architectures toward parallel processing as a means of providing cost-effective high performance computing environments (Myers, 1993; Pancake, 1991). Though parallel architectures take many forms, a basic distinction can be drawn on the basis of the number of instructions that are executed in parallel. A single instruction, multiple data (SIMD) stream computer executes the same instruction on several (often thousands of) data items in lock-step. This is often referred to as synchronous, fine-grained parallelism. A multiple instruction, multiple data (MIMD) stream computer, on the other hand, handles the partitioning of work among processors in a more flexible way, since processors can be allocated tasks that vary in size. Thus, programmers might assign portions of a large loop to different processors, or they might assign a copy of an entire procedure to each processor and pass a subset of data to each one. This allocation of parallel processes can occur in an architecture explicitly designed for parallel processing, or it may take place on a loosely-confederated set of networked workstations using software such as Linda (Carriero and Gelernter, 1990) or PVM (Beguelin *et al.*, 1991; 1993). Because of the flexibility associated with this coarse-grained MIMD approach, however, programmers must be concerned with balancing workloads across different processors. If a given processor finishes with its assigned task and it requires data being computed by another processor to continue, then a dependency is said to exist, the pool of available processors is being used inefficiently and parallel performance will be degraded (Armstrong and Densham, 1992).

The parallel computations described in this paper were performed using an Encore Multimax computer. The Encore is a modular MIMD computing environment with 32 Mbytes of fast shared memory; users can access between 1 and 14 NS32332 processors, and each processor has a 32K byte cache of fast static RAM. The 14 processors are connected by a Nanobus with a sustained bandwidth of 100 megabytes per second (ANL, 1987). Because a MIMD architecture is used in this research, workload distribution and dependency relations must be considered during the design of the interpolation algorithm.

IMPLEMENTATION OF THE ALGORITHM

The underlying assumption of inverse-distance weighted interpolation is that of positive spatial autocorrelation (Cromley, 1992): The contribution of near points to the unknown value at a location is greater than that of distant points. This assumption is embedded in the following equation:

$$z_j = \frac{\sum_{i=1}^{N} w_{ij} z_i}{\sum_{i=1}^{N} w_{ij}}$$

where:

z_j is the estimated value at location j,
z_i is the known value at location i, and
w_{ij} is the weight that controls the effect of other points on the calculation of z_j.

It is a common practice to set w_{ij} equal to d_{ij}^{-a}, where d_{ij} is some measure of distance and a is often set at one or two. As the value of the exponent increases, close data points contribute a greater proportion to the value of each interpolated cell (MacDougall, 1976:110; Mitchell, 1977: 256).

In this formulation, all points with known values (z_i) would contribute to the calculation of z_j. Given the assumption of positive spatial autocorrelation, however, it is common to restrict computations to some neighborhood of z_j. This is often done by setting an upper limit on the number of points used to compute the z_j values. The now-ancient SYMAP algorithm (Shepard, 1968), for example, attempts to ensure that between 4 and 10 data points are used (Monmonier, 1982) by varying the search radius about each z_j. If fewer than 4 points are found within an initial radius, the radius is expanded; if too many (e.g. >10) points are found, the radius is contracted. MacDougall (1976) also implements a similar approach to neighborhood specification. This process, while conceptually rational, involves considerable computation, since for each grid point, the distance between it and all control points must be evaluated.

Our parallel implementation of MacDougall's serial interpolation algorithm uses the Encore Parallel Fortran (EPF) compiler. EPF is a parallel programming superset of Fortran77 that supports parallel task creation and control, including memory access and task synchronization (Encore, 1988; Brawer, 1989).

Parallel Task Creation. An EPF program is a conventional Fortran77 program in which parallel regions are inserted. Such regions consist of a sequence of statements enclosed by the keywords **PARALLEL** and **END PARALLEL** and other EPF constructs can only be used within these parallel regions. Within a parallel region, the program is able to initiate additional tasks and execute them in parallel on the set of available processors. For example, the **DO ALL** ... **END DOALL** construct partitions loop iterations among the set of available processors. The number of parallel tasks, *n*, is specified by setting a processing environment variable called *EPR_PROCS*. At the command line, under the *csh* shell for example,

if one were to specify *setenv EPR_PROCS 4*, then three additional tasks would be created when the **PARALLEL** statement is executed.

Shared Memory. By default, all variables are shared among the set of parallel tasks unless they are explicitly re-declared inside a parallel region. A variable declared, or re-declared, inside a parallel region cannot be accessed outside that parallel region; each task thus has its own private copy of that variable. This behavior can be explicitly stressed by using the **PRIVATE** keyword when declaring variables. This approach could be used, for example, in a case when each of several (private) parallel tasks are required to modify specific elements in a shared data structure.

Process Synchronization. Portions of a program often require results calculated elsewhere in the program before additional computations can be made. Because of such dependencies, most parallel languages provide functions that allow the programmer to control the execution of tasks to prevent them from proceeding until they are synchronized with the completion of other, related tasks. In EPF, several synchronization constructs are allowed. For example, **CRITICAL SECTION** and **END CRITICAL SECTION** constructs enclose a group of statements if there is contention between tasks, so that only one task is allowed to execute within the defined block of statements and processing only proceeds when all tasks before the start of the critical section have been completed.

These parallel extensions facilitate the translation of serial code into parallel versions. In many instances, problems may need to be completely restructured to achieve an efficient parallel implementation, while in other instances conversion is much more straightforward. The serial to parallel conversion process often takes place in a series of steps, beginning with an analysis of dependencies among program components that may preclude efficient implementation (Armstrong and Densham, 1992). As the code is broken into discrete parts, each is treated as an independent process that can be executed concurrently on different processors. In this case, the interpolation algorithm can be cleanly divided into a set of independent processes using a coarse-grained approach to parallelism in which the computation of an interpolated value for each grid cell in the lattice is treated independently from the computation of values for all other cells. While some variables are declared private to the computation of each cell value, the matrix that contains the interpolated grid values is shared. Thus, each process contributes to the computation of the grid by independently writing its results to the matrix held in shared memory. In principle, therefore, as larger numbers of processes execute concurrently, the total time required to calculate results should decrease.

The following pseudocode, based on that provided in MacDougall (1984), presents a simplified view of a brute-force interpolation algorithm that uses several EPF constructs. The main parallel section, which calculates values for each grid cell, is enclosed between the DOALL ... END DOALL statements.

Code fragment 1. Pseudocode for parallel interpolation

```
DECLARATIONS
FORMAT STATEMENTS
FILE MANAGEMENT
READ DATA POINTS
CALCULATE INTERPOLATION PARAMETERS
    t_start = etime(tmp)
    FOR EACH CELL IN THE MAP
    PARALLEL
      INTEGER I, J, K, rad
      REAL   TEMP, T, B
      REAL   distance(Maxcoords)
      INTEGER L(Maxcoords)
      PRIVATE I, J, K, RAD, TEMP, T, B, distance, L
      DOALL (J=1:Columns)
        DO 710 I=1, Rows
           FOR EACH DATA POINT
           COMPUTE DISTANCE FROM POINT TO GRID CELL
           CHECK NUMBER OF POINTS WITHIN RADIUS
           COMPUTE INTERPOLATED VALUE
710   CONTINUE
      END DOALL
    END PARALLEL
   t_end = etime(tmp)
   t1 = (t_end - t_start)
 END
```

RESULTS

Because the Encore is a multi-user system, minor irregularities in execution times can arise from a variety of causes. To control the effects that multiple users have on overall system performance, we attempted to make timing runs during low-use periods. Given the amount of computation time required for the experiments using one and two processors, however, we did encounter some periods of heavier system use. These factors, however, typically cause only small irregularities in timings, thus the results reported in this paper are indicative of the performance improvements that can generally be expected in a MIMD parallel implementation.

The 10,000 point interpolation problem presents a formidable computational task that is well illustrated by the results in Table 3. When a single Encore processor is used, 33.3 hours is required to compute a solution to the problem. The run time is reduced to 2.5 hours, however, when all 14 processors are used. Figure 1 represents the monotonic decrease of run times with the addition of processors. Though the slope of the curve begins to flatten, a greater than order of magnitude decrease in computation time is achieved. This indicates that the problem is amenable to parallelism and that further investigation of the problem is warranted.

The results of parallel implementations are often evaluated by comparing parallel timing results to those obtained using a version of the code that uses a single processor. Speedup (see e.g. Brawer, 1989: 75) is the ratio of these two execution times:

$$\text{Speedup} = \frac{TimeSequential}{TimeParallel}$$

Measures of speedup can be used to determine whether additional processors are being used effectively by a program. The maximum speedup attainable is equal to the number of processors used, but speedups are normally less than this because of inefficiencies that results from computational overhead, such as the establishment of parallel processes, and because of inter-processor communication and dependency bottlenecks. Figure 2 shows the speedups obtained for the 10,000 point interpolation problem. The near-linear increase indicates that the problem scales well in this MIMD environment.

Table 3. Run times for the 10,000 point interpolation problem as different numbers of processors are used.

Processors	1	2	4	6	8	10	12	14
Hours	33.3	17.0	8.5	5.7	4.2	3.4	2.8	2.5

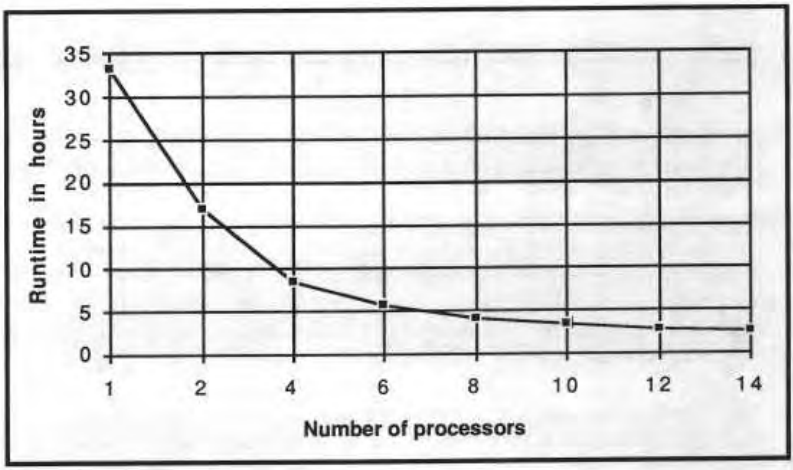


Figure 1. Run times for the 10,000 point problem.

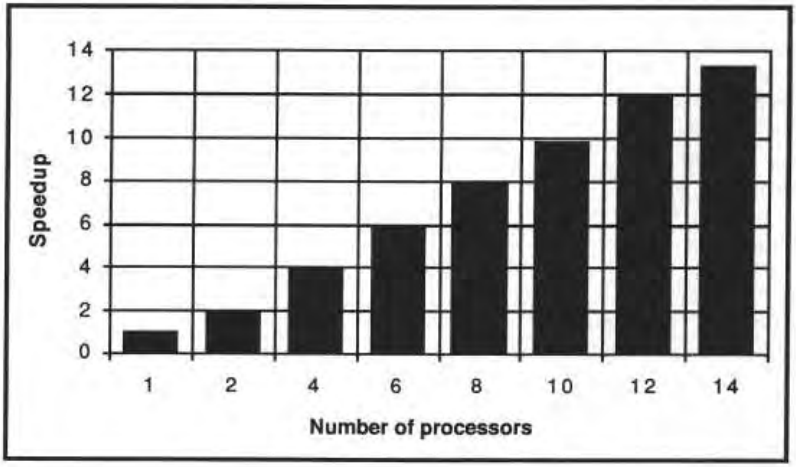


Figure 2. Speedups for the 10,000 point problem.

A measure of efficiency (Carriero and Gelernter, 1990: 74) is also sometimes used to evaluate the way in which processors are used by a program. This measure simply controls for the size of a speedup by dividing it by the number of processors used. If values start near 1.0 and remain there as additional processors are used to compute solutions, then the program scales well. On the other hand, if efficiency values begin to decrease as additional processors are used, then they are being used ineffectively and an alternative approach to decomposing the problem might be pursued. Table 4 shows the computational efficiency for the set of interpolation experiments. The results demonstrate that the processors are being used efficiently across the entire range of processors, with no marked decreases exhibited. The small fluctuations observed can be attributed to the lack of precision of the timing utility (etime) and random variations in the performance of a multi-processor, multi-user system. It is interesting to note, however, that the largest decrease in efficiency occurs as the number of processors is increased from 12 to 14 (the maximum). Because additional processors are unavailable, it cannot be determined if this decrease is caused by the presence of overhead that is only encountered as larger numbers of processors are added to the computation of results.

Table 4. Efficiency of the interpolation experiments.

Processors	2	4	6	8	10	12	14
Efficiency	0.99	0.99	0.97	0.99	0.98	0.99	0.95

CONCLUSIONS

Different approaches to improving the performance of inverse distance weighted interpolation algorithms have been developed. One successful approach (Hodgson, 1989) focused on reducing the total number of computations made by efficiently determining those points that are in the neighborhood of each interpolated point. When traditional, serial computers are used, this approach yields substantial reductions in computation times. The method developed in this paper takes a different tack by decomposing the interpolation problem into a set of sub-problems that can be executed concurrently on a MIMD computer. This general approach to reducing computation times will become increasingly commonplace since increasing numbers of computer manufacturers have begun to use parallelism to provide users with cost-effective high performance computing environments. The approach described here should also work when applied to other computationally intensive methods of interpolation such as kriging (Oliver *et al.*, 1989a; 1989b) that may not be directly amenable to the neighborhood search method developed by Hodgson.

Each processor in the MIMD computer that was used to compute these results is not especially fast by today's standards. In fact, when a modern superscalar workstation (RS/6000-550) was used to compute results for the same problem, it was 4.5 times faster than a single Encore processor. When the full complement of 14 processors is used, however, the advantage of the parallel approach is clearly demonstrated: the Encore is three times faster than the RS/6000-550. The approach presented here scales well with near linear speedups observed in the range from 2 to 14 processors.

Future research in this area should take place in two veins. The first is to use a more massive approach to parallelism. Because of the drop in efficiency observed when the maximum number of processors is used, larger MIMD-like machines, such as a KSR-1, or alternative architectures such as a heterogeneous network of workstations (Carriero and Gelernter, 1990) or a SIMD machine could be fruitfully investigated. It may be that a highly parallel brute force approach can yield performance that is comparable, or superior, to the search-based approaches suggested by Hodgson. The second, and probably ultimately more productive, line of inquiry would meld the work begun here with that of the neighborhood search-based work. The combination of both approaches should yield highly effective results that will transform large, nearly intractable spatial interpolation problems into those that can be solved in seconds.

ACKNOWLEDGMENTS

Partial support for this research was provided by the National Science Foundation (SES-9024278). We would like to thank The University of Iowa for providing access to the Encore Multimax computer and the Cornell National Supercomputing Facility for providing access to the RS/6000-550 workstation. Dan Dwyer, Smart Node Program Coordinator at CNSF, was especially helpful. Claire E. Pavlik, Demetrius Rokos and Gerry Rushton provided helpful comments on an earlier draft of this paper.

REFERENCES

ANL. 1987. *Using the Encore Multimax*. Technical Memorandum ANL/MCS-TM-65. Argonne, IL: Argonne National Laboratory.

Armstrong, M.P. and Densham, P.J. 1992. Domain decomposition for parallel processing of spatial problems. *Computers, Environment and Urban Systems*, **16** (6): 497-513.

Beguelin, A., Dongarra, J., Geist, A., Manchek, B., and Sunderam, V. 1991. *A User's Guide to PVM: Parallel Virtual Machine*. Technical Report ORNL/TM-11826. Oak Ridge, TN: Oak Ridge National Laboratory.

Beguelin, A., Dongarra, J., Geist, A., and Sunderam, V. 1993. Visualization and debugging in a heterogeneous environment. *IEEE Computer*, **26**(6): 88-95.

Brawer, S. 1989. *Introduction to Parallel Programming*. San Diego, CA: Academic Press.

Burrough, P.A. 1986. *Principles of Geographical Information Systems for Land Resources Assessment*. New York, NY: Oxford University Press.

Carriero, N. and Gelernter, D. 1990. *How to Write Parallel Programs: A First Course*. Cambridge, MA: The MIT Press.

Clarke, K.C. 1990. *Analytical and Computer Cartography*. Englewood Cliffs, NJ: Prentice-Hall.

Cromley, R.G. 1992. *Digital Cartography*. Englewood Cliffs, NJ: Prentice-Hall.

Encore. 1988. *Encore Parallel Fortran*, EPF 724-06785, Revision. A. Malboro, MA: Encore Computer Corporation.

Freund, R.F. and Siegel, H.J. 1993. Heterogeneous processing. *IEEE Computer*, **26**(6): 13-17.

Hodgson, M.E. 1989. Searching methods for rapid grid interpolation. *The Professional Geographer*, **41** (1): 51-61.

Lam, N-S. 1983. Spatial interpolation methods: A review. *The American Cartographer*, **2**: 129-149.

MacDougall, E.B. 1976. *Computer Programming for Spatial Problems*. London, UK: Edward Arnold.

MacDougall, E.B. 1984. Surface mapping with weighted averages in a microcomputer. *Spatial Algorithms for Processing Land Data with a Microcomputer*. Lincoln Institute Monograph #84-2. Cambridge, MA: Lincoln Institute of Land Policy.

Mitchell, W.J. 1977. *Computer-Aided Architectural Design*. New York, NY: Van Nostrand Reinhold.

Monmonier, M.S. 1982. *Computer-Assisted Cartography: Principles and Prospects*. Englewood Cliffs, NJ: Prentice-Hall.

Myers, W. 1993. Supercomputing 92 reaches down to the workstation. *IEEE Computer*, **26** (1): 113-117.

Oliver, M., Webster, R. and Gerrard, J. 1989a. Geostatistics in physical geography. Part I: theory. *Transactions of the Institute of British Geographers*, **14**: 259-269.

Oliver, M., Webster, R. and Gerrard, J. 1989b. Geostatistics in physical geography. Part II: applications. *Transactions of the Institute of British Geographers*, **14**: 270-286.

Pancake, C.M. 1991. Software support for parallel computing: where are we headed? *Communications of the Association for Computing Machinery*, **34** (11): 52-64.

Shepard, D. 1968. A two dimensional interpolation function for irregularly spaced data. *Harvard Papers in Theoretical Geography*, Geography and the Property of Surfaces Series, No. 15. Cambridge, MA: Harvard University Laboratory for Computer Graphics and Spatial Analysis.

White, D. 1984. Comments on surface mapping with weighted averages in a microcomputer. *Spatial Algorithms for Processing Land Data with a Microcomputer*. Lincoln Institute Monograph #84-2. Cambridge, MA: Lincoln Institute of Land Policy.

Implementing GIS Procedures on Parallel Computers: A Case Study

James E. Mower
Department of Geography and Planning
147 Social Sciences
State University of New York at Albany
Albany, New York 12222
jmower@itchy.geog.albany.edu

ABSTRACT

The development of efficient GIS applications on parallel computers requires an understanding of machine architectures, compilers, and message passing libraries. In this paper, the author compares performance statistics for implementations of drainage basin modeling, hill shading, name placement, and line simplification procedures under control-parallel and data-parallel approaches. The suitability of each approach and its message passing strategies are discussed for each application. The paper concludes with caveats for the developer concerning the current state of parallel programming environments.

INTRODUCTION

Parallel computing tools are being increasingly exploited by the GIS community for tasks ranging from line intersection detection (Hopkins, S., R.G. Healey, and T.C. Waugh 1992) to the development of network search algorithms (Ding, Y., P.J. Densham, and M.P. Armstrong 1992). GIS professionals who turn to parallel computers for fast solutions to large computing problems are confronted with an array of architectural designs not encountered in sequential computing. Choosing an inappropriate parallel architecture or computing model can result in little or no performance benefits over a modern sequential computer. This paper will illustrate the suitability of various parallel computing models for GIS applications by examining the performance characteristics of code written by the author for:

1) drainage basin analysis,
2) analytical hill shading,
3) cartographic name placement, and
4) line simplification.

It will show that the appropriate selection of a parallel computing architecture and programming environment is essential to the visualization and efficient solution of the problem.

The differing computational demands of each problem domain highlight the strengths of competing parallel architectures. The name placement procedure, making extensive use of general interprocessor communication services, is contrasted with the hill shading procedure that uses simple grid communication facilities. Two procedures based on the Douglas line simplification

algorithm (Douglas and Peucker 1973), one written as a control-parallel procedure and the other as a data-parallel procedure, show the relative costs of message passing on multiple instruction stream, multiple data stream (MIMD) computers and on single instruction stream, multiple data stream (SIMD) computers.

The procedures described in this paper were implemented on Thinking Machines CM-2 and CM-5 computers. The CM-2 is a true SIMD computer, consisting of up to 64K bit-serial processors connected by hypercube and grid communication networks. The CM-5 is a synchronous-asynchronous multiple data (SAMD) computer, capable of operating in both control-parallel and data-parallel modes. The name placement procedure was implemented on a CM-2; all the other procedures were implemented on a CM-5.

Where appropriate, performance statistics are provided to show the consequences of making specific design choices. Particular emphasis will be given to the performance of the control-parallel and data-parallel variants of the Douglas algorithm, both running on the CM-5.

If a programming environment lacks a structure for stating clear solutions to a class of problems, attaining marginal increases in performance may be inconsequential to the developer. Comparisons of the Douglas implementations will show that the control-parallel model better captures the elegance of the sequential algorithm.

MACHINE ARCHITECTURES AND PROGRAMMING MODELS

This paper will examine the procedures in the context of SIMD and MIMD architectures. SIMD machines support data-parallel programming. Under this approach, a control or front-end processor broadcasts instructions to a parallel processor array. All processors in the array operate synchronously, each executing or ignoring the current instruction, depending upon the state of its local data. The data-parallel procedures described here are implemented in the C* programming language, a superset of ANSI C with data-parallel extensions.

MIMD machines allow processors to run asynchronously on their own instruction stream. In a master-worker control-parallel model, each worker processor runs an identical copy of a program as if each were a separate computer, exchanging data with the master through message passing operations. Because the instruction streams are independent, the rate at which each worker executes its instructions is determined by the length of its data stream (Smith 1993).

Workers notify the master processor when they are ready to receive new work or when they are ready to send completed work. During a cooperative or synchronous message passing operation, the receiving processor remains idle until the sending processor is ready to pass its message.

IMPLEMENTING THE PROCEDURES

Drainage basin modeling and hill shading

The data-parallel drainage basin and hill shading procedures map elevation samples in a U.S.G.S. 1:24,000 series DEM to processors arranged in a 2-dimensional grid. The procedures execute the following computations:

1) removal of most false pits through elevation smoothing,
2) calculation of slope and aspect to determine cell drainage direction,
3) propagation of drainage basin labels,
4) removal of remaining false pits through basin amalgamation,
5) location of stream channels through drainage accumulation, and
6) calculation of hill shaded value from cell slope and aspect and light source azimuth and altitude.

To take full advantage of a SIMD machine, a data-parallel program must execute instructions on all of the array processors, and hence on all of its input simultaneously. For drainage basin analysis and hill shading, Mower (1992) shows that steps 1, 2, and 6 meet this criterion. Typically, though, many data-parallel procedures operate on a subset of the processors over any particular instruction. Steps 3, 4, and 5 all require that values propagate away from a set of starting cells. In steps 3 and 4, drainage basin labels propagate away from pits. On the first iteration, only processors associated with pits are active. On subsequent iterations, progressively larger numbers of processors are activated along the 'wave front' expanding away from each pit; cells that were active on the previous iteration become inactive (Figure 1).

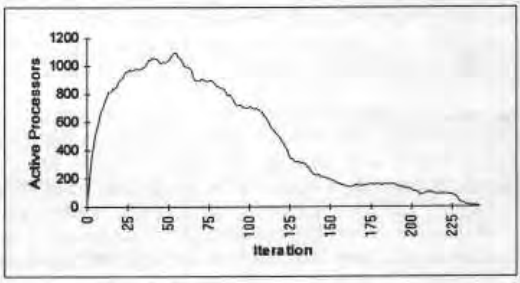


Figure 1. Processors that are active over each iteration of step 3 for a 1:24,000 DEM sampled at 30 meter intervals. The entire DEM is represented by approximately 120,000 processors.

Step 5 promotes an opposite pattern of activation. On the first iteration, each cell starts with one unit of water and queries its 8-case neighbors for cells that drain toward itself. After accumulating the water supplies of

the uphill neighbors, the next iteration begins. If a cell no longer finds uphill neighbors with a positive water supply, it becomes inactive. The number of active processors gradually declines until the end of the step when none but those associated with pits are active (Figure 2).

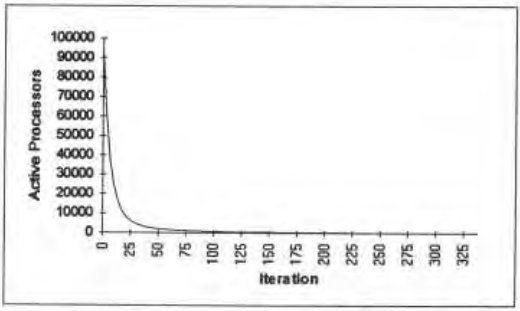


Figure 2. Active processors over each iteration of the drainage accumulation procedure for the same size window as Figure 1.

Steps 1, 2, and 6 of the data-parallel algorithms for drainage basin modeling and hill shading are relatively simple to implement on a SIMD machine. They are also fast—hill shading an entire 1:24,000 DEM requires less than one second. For each step, the iterative control loop of a sequential implementation is replaced with a parallel context operator that restricts the computations to all but the edge cells of the matrix. Operations that require values from 8-connected neighbors receive them through fast grid communication functions. Steps 3, 4, and 5 are harder to implement, requiring that processors determine their state of activation with respect to their local data.

Operations that refer to a data object larger than the grid cell are sometimes clumsy or inefficient to implement. To find the pour point for a drainage basin in step 4, cells on the edge of the basin are activated. A scanning function finds the pour point as the activated cell with minimum elevation that also drains to a basin with a lower pit than its own. Finally, cells in the basin with elevations below the pour point are activated to raise them to the level of the pour point in a flooding procedure. These operations pertain to a small percentage of the total cells in the DEM, leaving the rest of the processors inactive over their duration.

The author is currently developing a control-parallel algorithm for drainage basin modeling, using the drainage basin, rather than the grid cell, as the basic data object. This approach applies a standard suite of drainage basin modeling procedures to each basin as defined initially by its pit. Using a master-worker model, the master processor assigns pits to workers as they become available. Each worker runs asynchronously on

its basin, requesting a new basin from the master on completion.

This approach will perform best on large regions containing many basins. For smaller regions having fewer basins than processors, dissimilarities in running time among the workers will lead to relatively low efficiency. This approach also lacks the elegance of the data-parallel procedure with regard to steps 1, 2, and 6, ignoring their inherent parallel structures.

Name placement

The name placement procedure employs a data-parallel model equating processors with the locations of populated places. (Mower 1993a). These features are distributed irregularly across the earth and cannot be represented efficiently in a two-dimensional array. Instead, their processors are represented as a vector. Since the position of a processor in the vector array does not implicitly represent the geographic coordinates of its place, the coordinates must be stored explicitly.

As the procedure scans the database for places that fall within the user's window, it determines the neighborhood for each place as the total area over which its label could be located. Places later compare their neighborhoods against one another for intersection. If a place finds an overlapping neighborhood, it adds the processor identifier of the overlapping place to its list of neighbors. Currently, data is read from modified U.S.G.S. Geographic Names Information System (GNIS) files of populated places for names appearing on 1:24,000 series maps.

For its point symbol and label to occupy non-overlapping map locations, each place queries the locations of the point markers and labels of its neighbors. Since the geographic location of each place is arbitrary with respect to the position of its processor in the list, interprocessor communication must occur over a general communication network. Depending upon the topology of the network and upon the relative addresses of the processors, messages will require varying numbers of steps to reach their destination. If messages contend for a limited number of network router nodes, the overall time to pass them will increase. Therefore, general communication functions frequently take longer to perform than do grid functions, usually requiring one step.

After querying the user for an input window and map scale, the name placement procedure begins by placing all point features (currently populated places) that fall in the user's window onto the map with their labels to the upper right of their point symbols. Depending on map scale, feature density, feature importance, type size, and point symbol size, a number of feature symbols and labels will overlap. Each processor checks the features in its neighborhood for overlap conditions. If it finds that its label overlaps one or more features of greater importance

(currently determined as a simple function of population), it tries moving its own label to an empty position. If no such position exists, it tries moving its label to a position that is occupied by no features of greater importance than itself. If that doesn't work, the feature deletes itself. All features iterate through this cycle until no conflicts remain.

The pattern of processor activation for name placement is quite different from those for drainage basin analysis and hill shading which are generally similar across data sets. The running time of the name placement program varies directly with the maximum number of neighbors found by any processor. This is a function of map scale—at some very large scale, no neighborhoods overlap; at some very small scale, all neighborhoods overlap. As map scale decreases, the maximum number of neighbors found by any map feature increases, requiring the activation of a greater number of processors. The number of active processors and the lengths of the neighbor lists decline as overlaps are resolved through label movement or feature deletion, increasing the execution speed of subsequent iterations.

The author compared the performance of the SIMD version implemented on a CM-2 to a functionally equivalent sequential version running on a Sun Microsystems SPARCstation 2. For a 1:3,000,000 scale map of New York State, the SPARCstation required one hour and 25 minutes. The same map completed in slightly over 4 minutes on the CM-2. The author also found that the running time of the CM-2 version increased linearly at a small constant rate with the maximum number of neighbors (as a function of map scale) of any mapped feature. With a 20 fold increase in the maximum number of neighbors, running time increased by a factor of only 2.15 (Figure 3).

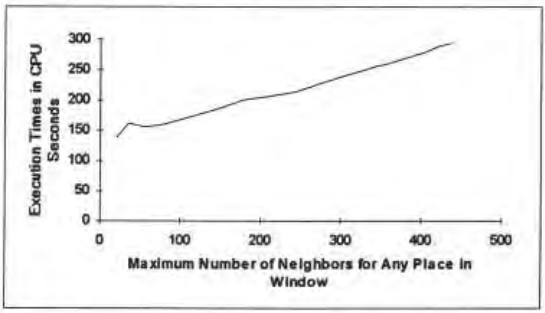


Figure 3. Graph of execution time as a function of the maximum number of neighbors for any place on the map.

Line simplification

The Douglas algorithm for line simplification is implemented in both data-parallel and control-parallel approaches. Under the data-parallel approach, each processor represents the vertex of a cartographic line, extracted from U.S.G.S. 1:2,000,000 series DLG files.

Under the control-parallel approach, copies of a simplification procedure run asynchronously on each processor. Two versions of this approach are implemented. In the first version, each processor is responsible for opening a line file, reading its contents, simplifying all lines in the file within the user's window, and writing the output to a global file. In the second version, a master processor performs all input and distributes work, segmented by lines, to the next available worker processor that requests it. Each worker simplifies its line and writes its output to a global file. The master and the workers negotiate the distribution of work and data through synchronous (cooperative) message passing.

Under the data-parallel approach, the set of processors dedicated to a line are divided into scan sets. A scan set consists of processors representing the starting node, the ending node, and the vertices between them. A processor representing a starting node finds the equation of the baseline connecting itself to the ending node. The vertex processors calculate their perpendicular distances to the baseline. A scanning function on the front-end finds the vertex processor in each scan set having the greatest calculated distance from its baseline, marking it significant if the distance is greater than the user's tolerance value. Each significant vertex segments the line into new scan sets. The procedure continues until it finds no new significant vertices.

Of the three approaches to line simplification, the first is the easiest to implement. It simply applies a recursive sequential implementation of the Douglas procedure separately to each file that happens to intersect the user's window. No message passing is required between the master processor and a worker once its file is opened and processing begins. For each file, the number of lines per unit area and the amount of overlap between its region and that of the user's window determine the length of its processing time. Given an even distribution of lines across the region, files overlapping the edge of the user window will finish sooner than those in the middle of the window.

The second approach uses a master-worker strategy to balance the workload more evenly across the processors. It also runs a recursive sequential implementation on each node but segments the data by line rather than by file. The master processor polls the workers for any that are idle, sending each available worker an individual line to simplify. When a worker has completed its line, it writes it to a global output file.

The second approach requires 2 types of messages to be sent: 1) a message from a worker announcing that it is ready to receive a line for processing and 2) a message from the master sending a worker a line to simplify. Both messages are passed cooperatively, requiring the receiving partner to remain idle until the message originates from the sending partner.

Mower (1993b) compares execution times for the data-parallel and control-parallel approaches to the Douglas algorithm. The parallel approaches were compared to a sequential version built from the control-parallel-by-file code. To control for differences in operating system performance, the sequential version was linked to the same parallel libraries as the control-parallel versions but run on a single CM-5 processor.

Surprisingly, the control-parallel version segmented by files and the sequential version both performed substantially faster than the version segmented by lines. Although the latter provides better load balancing, it does so through expensive message passing operations. Figure 4 compares the amount of time required to execute each version on varying numbers of lines. Except for message passing and I/O operations, the source code for the three versions are identical.

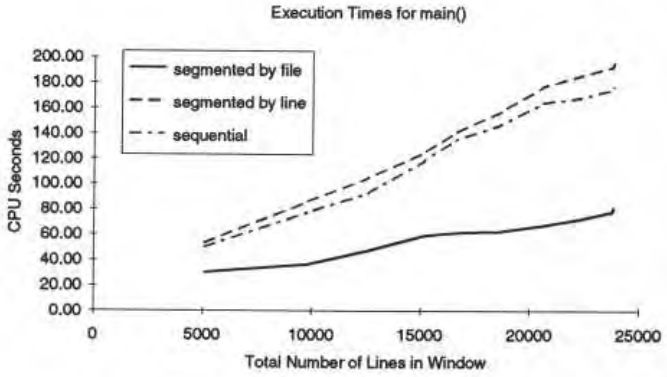


Figure 4. Graph of execution times for each of three parallel implementations of the Douglas Procedure.

Because test results of a data-parallel prototype implementation gave very slow execution times compared to the control-parallel prototypes, it was not developed to a final testing version. The cause for its poor performance can be attributed mainly to its inefficient use of SIMD processors. The prototype implementation processes one line at a time, leaving the majority of the processors idle. The procedure would activate a greater number of processors if more lines were processed simultaneously but would still perform the same number of sequential scans over each scan set. It is unclear whether this would lead to a substantial improvement in its performance characteristics.

CONCLUSION

This paper has shown the performance benefits and liabilities incurred through the application of various parallel architectures, programming approaches, and

message passing strategies to drainage basin modeling, hill shading, name placement, and line simplification implementations. Specifically, it found that:

1) data-parallel procedures execute fast when all processors are kept active and message passing is restricted to grid operations;

2) data-parallel procedures may perform slower than equivalent sequential procedures when the conditions in 1) are not met;

3) synchronous message passing operations slow execution substantially in control-parallel procedures; and

4) of the procedures reviewed in this paper, name placement and hill shading currently offer the best performance improvements over equivalent sequential procedures.

Some observations on parallel programming

SAMD machines provide the necessary flexibility for implementing the procedures described in this paper, offering control-parallel and data-parallel programming models under a variety of message passing schemes. At the moment, however, some of these tools work better than others. The author's experience in the development of parallel procedures for GIS applications on the CM-5 has led to the following observations on its programming environments:

1) Parallel programming languages and compilers are changing rapidly. Modern parallel computers support parallel versions of several high-level programming languages including C, FORTRAN, and Lisp. A manufacturer will generally supply new compilers with new or updated platforms. Changes from one version of a compiler to the next may require the user to spend many hours debugging or optimizing code for the new environment. The author has found that the best defense against compiler changes is to use the simplest data structures and control structures that the language offers. Unfortunately, many of the new compilers are actually beta test versions and do not perform good code optimization. In that case, the user must select those language features that are known to compile into the fastest executable code. Expect to rewrite your code frequently.

Sometimes languages themselves change. Early versions of the C* programming language implemented by Thinking Machines, Inc. looked very much like C++. After version 6, the language was completely rewritten and now looks much more like standard C. Most of the parallel extensions changed completely between versions.

2) Debugging is difficult. Data-parallel debuggers have become much more usable with the introduction of X-Windows-based data visualization tools. Some of these

tools are still limited in their ability to show members of parallel structures or elements of parallel arrays. Traditional debugging methods using print statements from within a program may not be successful in printing values of parallel or local variables. As a result, the programmer must often resort to indirect techniques such as introducing scalar debugging variables within a program.

Control-parallel debugging with print statements is somewhat easier to perform since the processors run standard versions of sequential programming languages. In this environment, the programmer generally tries to reduce the amount of debugging output or to simplify it. On a MIMD machine of 32 processors, the output of a single print statement in straight-line code would appear 32 times, once for each copy of the program. For procedures with large numbers of print statements embedded within complex control structures, the output of the statements can be intermingled, making for confusing reading.

The relevance of these observations will fade as parallel programming environments stabilize. With an understanding of their current limitations, the GIS developer can use these tools to bring about large increases in the performance of spatial data handling procedures.

REFERENCES

Ding, Y., P.J. Densham, and M.P. Armstrong 1992, Parallel Processing for Network Analysis: Decomposing Shortest Path Algorithms for MIMD Computers, Proceedings, 5th International Symposium on Spatial Data Handling, Charleston, pp. 682-691.

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, The Canadian Cartographer, 10(2):112-122.

Hopkins, S., R.G. Healey, and T.C. Waugh 1992, Algorithm Scalability for Line Intersection Detection in Parallel Polygon Overlay, Proceedings, 5th International Symposium on Spatial Data Handling, Charleston, pp. 210-218.

Mower, J.E. 1992, Building a GIS for Parallel Computing Environments: Proceedings, 5th International Symposium on Spatial Data Handling, Charleston, pp. 219-229.

Mower J.E. 1993a, Automated Feature and Name Placement on Parallel Computers, Cartography and Geographic Information Systems, 20(2):69-82.

Mower J.E. 1993b (manuscript submitted for review), SIMD Algorithms for Drainage Basin Analysis.

Smith, J.R. 1993, The Design and Analysis of Parallel Algorithms, Oxford University Press, Oxford.

SUITABILITY OF TOPOLOGICAL DATA STRUCTURES FOR DATA PARALLEL OPERATIONS IN COMPUTER CARTOGRAPHY

Bin Li
Department of Geography
University of Miami
Coral Gables, FL 33124-2060
Internet: binli@umiami.miami.edu

ABSTRACT

High performance computing is rapidly evolving towards parallel processing. Among various issues in applying parallel processing to computer cartography, a major concern is the compatibility between parallel computational models and existing cartographic data structures as well as algorithms. This paper assesses the suitability of topological data structures for Data Parallel Processing in computer cartography. By mapping selected cartographic operations to the data parallel model, we found that topological data structures, with minor extensions, can support data parallel operations. The topological information stored in the cartographic database can be used as index vectors for segmented-scan and permutation. To demonstrate this approach, the paper describes data parallel implementations of three cartographic operations are described, including line generalization, point-in-polygon search, and connectivity matrix construction.

INTRODUCTION

The motivation to conduct this research is three-fold. First, recent studies have shown the great potential of data parallel processing in computer cartography and GIS (Mower, 1992, 1993; Mills et al., 1992a, 1992b; Li, 1993). While most studies have been conducted in the raster domain, data parallel processing in the vector space has received less attention. Due to their spatial irregularity, cartographic operations in the vector domain are more difficult to adopt to the data parallel computational model. To perform data parallel processing, the vector space must be transformed to a regular one through specific decompositions such as the *uniform grid* (Franklin et al., 1989; Fang and Piegl, 1993). Although the *uniform grid* is a very efficient data structure for many vector operations, one of its drawbacks is the overhead to convert the data from and to their original structures. The conversion is necessary in the real world situations because few software packages for GIS and computer cartography use the *uniform grid.* It is therefore worthwhile to study the suitability of existing vector data structures for data parallel processing. Topological data structures are selected in this project because they are the most commonly used data structures in GIS software packages.

Second, large scale applications of the parallel computing technology require "enabling technologies" that are based on existing hardware and software environment and allow flexible evolutions. Set aside other technical and economic conditions for using parallel computing, software developers would not adopt strategies that require fundamental changes in existing data

structures. It is preferable to have solutions that are less costly but can greatly increase processing capacity and functionality. One of such solutions is to develop add-on software that enables the user to run large GIS applications on parallel computers. Such add-on software must provide interface between the core modules and the new parallel programs. The interface will be more efficient if the existing data structure is compatible with the parallel computational model.

Third, it is conceptually challenging to examine how topological representations of geographic space may facilitate data parallel processing in cartography. Understanding such relations may help developing new insights to solving cartographic problems.

This paper reports findings on three popular operations in computer cartography and GIS. They are line simplification, point-in-polygon search, and connectivity matrix construction. The paper describes how data parallel procedures can be implemented with topological data structure. It emphasizes on whether the selected cartographic algorithms can be expressed with data parallel operations without significant alternation on the original data structure.

The topological data structure used in the paper is described in ESRI's GIS training book (ESRI, 1990). Some good references on data parallel processing include the article by Hillis et al. (1986), the book by Blelloch (1991), the FORTRAN 90 Handbook by Admas et al. (1992), and a number of programming manuals from Thinking Machines Corporation (1991a, 1991b).

LINE SIMPLIFICATION

Line simplification is an important operation in cartographic generalization. A commonly used procedure is the Douglas-Peucker algorithm (Douglas and Peucker, 1973). Mower (1992) presented strategies to implement the algorithm for the data parallel (S-LINE) and the message passing (M-LINE) computational model. However, Mower's parallel procedures seem to apply to only a single line, which may not be applicable to a vector coverage that has more than one line. This section describes a data parallel procedure that executes the Douglas-Peucker algorithm on all the lines simultaneously.

The basic strategy is to store the coordinates of all the points on several long vectors, with two start_bit vectors defining the boundaries between line segments. *Prefix scan* can then be used to find the significant points in all segments. The process iterates until all points are either retained or eliminated.

The operation requires eight vectors:

- vector X and Y for the original x, y coordinates,
- vector Xs, Ys, Xe, and Ye for the start and end point coordinates,
- vector DIST to store the distance from each point to the line that links the two end points in each line segment,
- vector FLAG to record the status of each point on the line.

These vectors are segmented with two start-bit vectors, **S_up** and **S_down**. They specify the boundaries of line segments from the upward (from left to right) or the downward (from right to left) direction (Figure 1). Because new significant points are generated in each iteration, the length of these vectors are allocated dynamically.

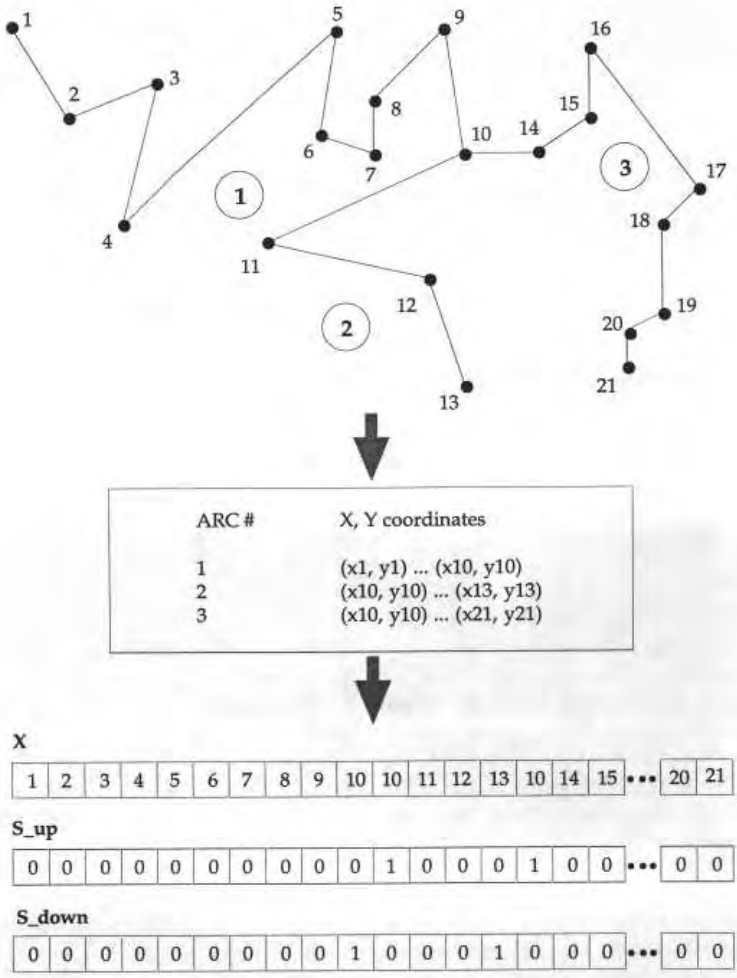


Figure 1. Mapping a line coverage to segmented vectors defined by two start-bits. Vector **X** stores the x coordinates of all the points. The numbers in **X** are the IDs of the points. Other vectors, including **Y, Xs, Ys, Xe, Ye, DIST,** and **FLAG,** have the same structures as **X**.

(1) Construct vector **X, Y, S_up,** and **S_down** from the arc-coordinate list. Generating vector **X** and **Y** requires a simple assignment but the start-bit vectors need to be constructed with parallel permutation. The parallel indices for the permutation are slight modifications of the running-totals of the vector "number-of-points" obtained from the arc-coordinate list. These

indices direct the assignment of scalar 1 to the start-bit vector **S_up** and **S_down** (Figure 2).

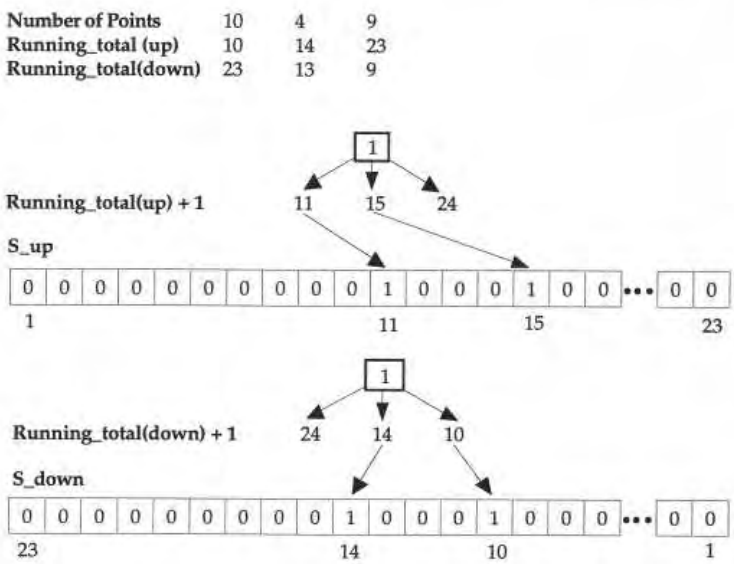


Figure 2. Constructing the start-bits from the "number-of-points" vector. Because *prefix-scan* with start-bit is direction dependent, two start-bit vectors are needed to define line segments.

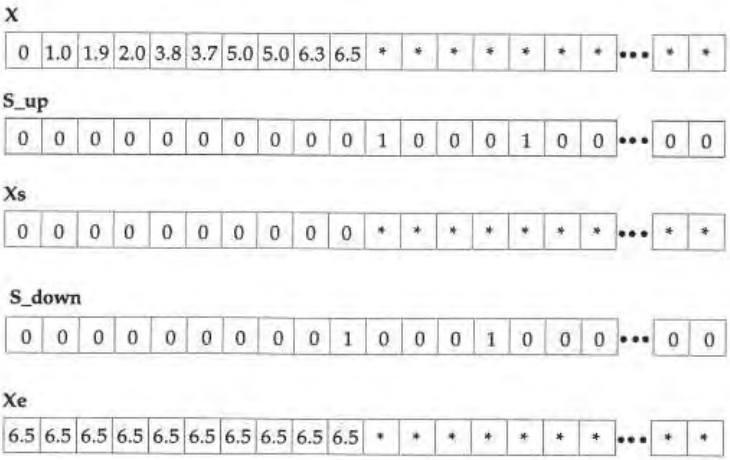


Figure 3. Illustration of the copy-scan operation. Only line segment 1 is shown here. Other line segments are marked with "*". Note that the x coordinate of the start point, 0, is distributed to all points in segment 1 of **Xs**; and the end point coordinate, 6.5, is copied to all locations in segment 1 of **Xe**.

(2) Assign 1 to positions in vector **FLAG** that correspond to the start and end points. By default, the start and the end points are significant and retained.

(3) Use *copy-scan* to distribute start/end point coordinates from vector **X** and **Y** to corresponding line segments in vector **Xs, Ys, Xe, Ye** (Figure 3).

(4) Calculate the distances from each point to the line that links the two end points of a corresponding line segment. This step involves element-wise operations among vector **X, Y, Xs, Ys, Xe,** and **Ye**. The results are stored in vector **DIST**.

(5) Use *maximum-scan* to find the longest distance in each segment in **DIST**. Select positions that are greater than the user-specified tolerance, record them as significant points in vector **FLAG**.

(6) Calculate the new lengthes of the vectors based on the number of new significant points found. Since a new significant point becomes a start point of one line segment as well as an end point of another, for each new significant point, one position must be added to the vectors. The calculation can be done with the selection procedure in step (5).

(7) Permute the original **X, Y** coordinates to the new vectors (Figure 4).

(8) Update the start-bit vectors so that locations corresponding to the new significant points are included as start-bits (Figure 4).

(9) Repeat (3) to (8) until all entries in vector **FLAG** become either 1 or 0. Retain points that have flag value 1.

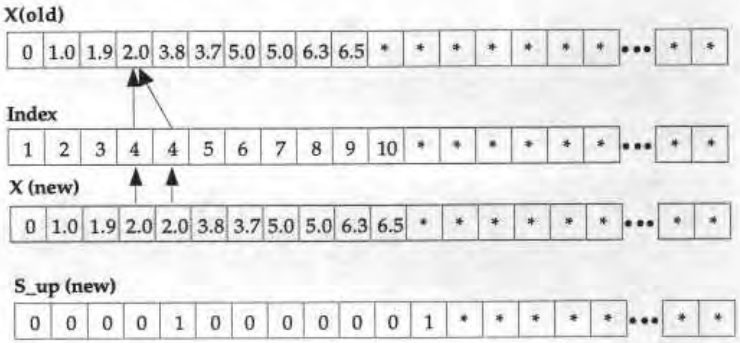


Figure 4. New vectors are created in each iteration. Vector **Index** directs **X(new)** to obtain coordinates in corresponding locations in **X(old)**. The start-bit vector is also updated. Note that the new vectors may have different length from the previous ones because line segments split at the new significant points.

POINT-IN-POLYGON SEARCH

Point-in-polygon search can be accomplished by the plumb-line algorithm—the search point (X0, Y0) is in polygon P if the number of intersects between the plumb-line X = X0 and P, and above the search point, is odd (Monmonier, 1982). Since intersects are calculated with each link (a line

segment defined by two points), it is necessary to establish the arc-link topology (Figure 5).

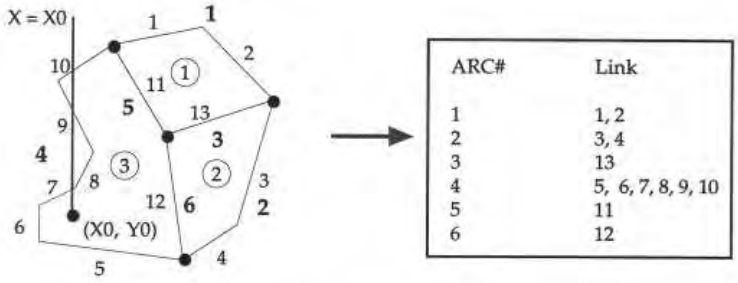


Figure 5. The polygon, the search point, and the arc-link topology.

Once the arc-link topology is built, the linkages between the links and the polygons also are established. We can first identify links that intersect with the plumb-line above the search point. Then with the topology vectors as parallel indices, the number of intersects can be permuted to arcs then to polygons.

(1) Find intersects between the links and the plumb-line. The following parallel structure is used to accommodate the calculations:

```
shape [N]vector;
struct segment {
            int    id;
            float  x1, y1;
            float  x2, y2;
            int    flag;
};
struct segment:vector     link;
```

illustrated as:

```
LINK    1  2  3  4  5  6  7  8  9  10  11  12  13
        X1 ...
        Y1 ...
        X2 ...
        Y2 ...
        flag ...
```

Each element in **LINK** stores the link ID, the coordinates of two endpoints, and a flag to indicate whether the link intersects with the plumb-line. To check intersects, the x, y coordinates of the search point are broadcasted to all locations in **LINK** and the calculations are performed simultaneously. Links that intersect with the plumb-line above the search point have flag value 1, others 0. For the example in figure 5, the intersect flags turn out as follows:

0 0 0 0 0 0 1 0 1 1 0 0 0

(2) Find the number of intersects with each arc (Figure 5).

• Use vector **Arc-link** as the parallel index, get the intersect flags to vector **Arc-intersect** so that the intersect information could be associated with each arc.

• Use vector **Arc-ID** as the parallel index, combine the number of intersects to vector **Arc**.

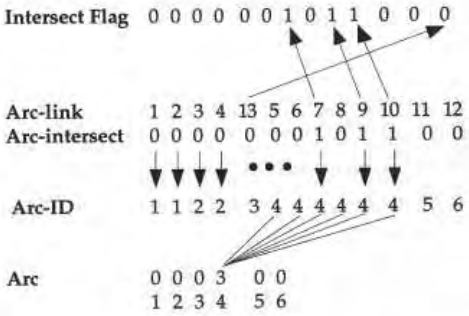


Figure 6. Using parallel communications to find the number of intersects with each arc.

(3) Find the number of intersects with each polygon. Similar to the arc-link topology, the polygon-arc relation is mapped to vector **Polygon-arc** and **Polygon-ID** (Figure 7). Then the same procedures in step 2 are used to combine the total number of intersects with each polygon (Figure 8):

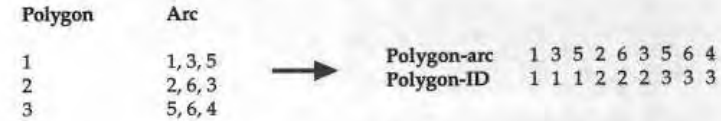


Figure 7. Deriving parallel index vectors from the polygon-arc topology.

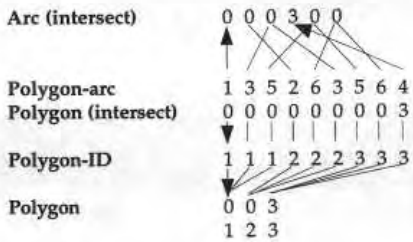


Figure 8. Using parallel communication to find the number of intersections with each polygon.

• Use vector **Polygon-arc** as the parallel index, identify which arc intersects with the plumb-line.

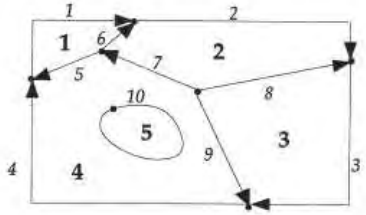
• Use vector **Polygon-id** as the parallel index, combine the number of

intersects to vector **Polygon**.

(4) Select from vector **Polygon** the location that has the odd number of intersects.

CONNECTIVITY MATRIX CONSTRUCTION

Connectivity matrix is commonly used in spatial statistics. It is a numerical representation of spatial relations for one and two dimensional cartographic objects. With cartography evolving beyond its traditional scope of mapping geographic space and relations, spatial connectivity matrix becomes a necessary component for inferential cartographic analysis.



Arc #	Left	Right
1	0	1
2	0	2
3	0	3
4	0	4
5	4	1
6	1	2
7	4	2
8	2	3
9	3	4
10	4	5

(a)

	0	1	2	3	4	5
0	0	1	1	1	1	0
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	0	1	1	0	0	1
5	0	0	0	0	0	0

(b)

	0	1	2	3	4	5
0	0	1	1	1	1	0
1	1	0	1	0	1	0
2	1	1	0	1	1	0
3	1	0	1	0	1	0
4	1	1	1	1	0	1
5	0	0	0	0	1	0

(c)

Figure 9. Constructing a connectivity matrix from the left-right topology.

The following describes how the left-right topology can accommodate data parallel construction of the connectivity matrix for a polygon coverage. The procedure is straight-forward—the two polygons that share the same arc are connected. In addition, an arc cannot be shared by more than two polygons. Therefore, given vector LEFT and RIGHT, LEFT(i) and RIGHT(i) are neighboring polygons (Figure 9a). In other words, the LEFT and RIGHT vector actually serve as the indices for the two dimensional connectivity matrix. Using the FORALL construct in FORTRAN90, the relations between the connectivity matrix and the LEFT and RIGHT vector can be expressed as:

```
FORALL (i = 1:N)  C_mat(LEFT(i), RIGHT(i)) = 1
```

where **C_mat** is an N-by-N binary matrix, with 0 and 1 indicating the connectivity between two polygons (Figure 9b). A complete **C_mat** is obtained by combining with its transpose, i.e.,

C_mat = TRANSPOSE(C_mat) + C_mat

where TRANSPOSE is an intrinsic function in FORTRAN90 (Figure 9c).

SUMMARY

This preliminary study found that the topological data structure sufficiently supports data parallel implementation of three cartographic operations. First, the topologies stored in the vector cartographic database can be used as parallel indices to establish hierarchical relations among cartographic objects. For instance, using the arc-link topology as the parallel index, an arc vector can be easily constructed from the link vector. Similarly, a polygon vector can be built with the polygon-arc topology. The following C* statement express the hierarchical relations among cartographic objects:

```
ARC = [ARC_LINK] LINK;
POLYGON = [POLY_ARC]ARC;
```

Such linkages also channel information among cartographic objects from one level to another. The section on point-in-polygon search showed how the number of intersects can be accumulated from links to polygons.

Second, vectors that define segments for parallel prefix scan operations are also generated from topological information in the original database. Segmented scan makes it possible to execute accumulative operations simultaneously on all units of such cartographic objects as arc and polygon. The boundaries between cartographic objects are defined with two approaches, using the start-bit vector or the ID vector. Both are generated from the original topological information. With these vectors, many cartographic algorithms can be implemented with segmented *prefix-scan* which we used as the primary operation for line simplification.

Findings from this study should be applicable to data parallel implementations of other vector-oriented operations in analytical cartography, such as polygon overlay and network analysis. They also should be useful for assessing the technical feasibility of data parallel processing in the vector domain.

ACKNOWLEDGEMENT

The author wishes to thank the Northeast Parallel Architecture Center at Syracuse University for providing the computer hardware and technical supports.

REFERENCES

Adams, J., et al., 1992, FORTRAN 90 Handbook, Complete ANSI/ISO Reference, McGraw-Hill Book Company, New York.

Blelloch, G., 1991, Vector Models for Data-Parallel Computing, MIT Press, Cambridge, MA.

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: The Canadian Cartographer, Vol. 10, No. 2, pp. 112-122.

ESRI, 1990, Understanding GIS, ESRI Inc., Redland, CA.

Fang, T. P., and L. Piegl, 1993, Delaunay Triangulation Using a Uniform Grid: IEEE Computer Graphics and Applications, Vol. 13, pp. 36-47.

Franklin, R., et al., 1989, Uniform Grids: A Technique for Intersection Detection on Serial and Parallel Machines: Auto Carto 9, Proceedings, Ninth International Symposium on Computer-Assisted Cartography, Baltimore, MD, pp. 100-109.

Hillis, D., and G. L. Steele, Jr., 1986, Data Parallel Algorithms: Communications of ACM, Vol. 29, pp. 1170-1183.

Li, Bin, 1993, Opportunities and Challenges of Parallel Processing in Spatial Data Analysis: Initial Experiments with Data Parallel Map Analysis, Doctoral Dissertation, Department of Geography, Syracuse University, Syracuse, NY.

Mills, K., et al., 1992a, Implementing an Intervisibility Analysis Model on a Parallel Computing System: Computers & Geosciences, Vol. 18, pp. 1047-1054.

Mills, K., et al., 1992b, GORDIUS: A Data Parallel Algorithm for Spatial Data Conversion: SCCS-310, Syracuse University, Syracuse, NY.

Monmonier, M., 1982, Computer Assisted Cartography: Principles and Prospects, Prentice Hall, Englewood Cliffs, NJ.

Mower, J., 1992, Building a GIS for Parallel Computing Environment: Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC, pp. 219-229.

Mower, J., 1993, Automated Feature and Name Placement on Parallel Computers: Cartography and Geographic Information Systems, Vol. 20, No. 2, pp. 69-82.

Thinking Machines Corporation, 1991a, Programming in FORTRAN, Cambridge, MA.

Thinking Machines Corporation, 1991b, Programming in C*, Cambridge, MA.