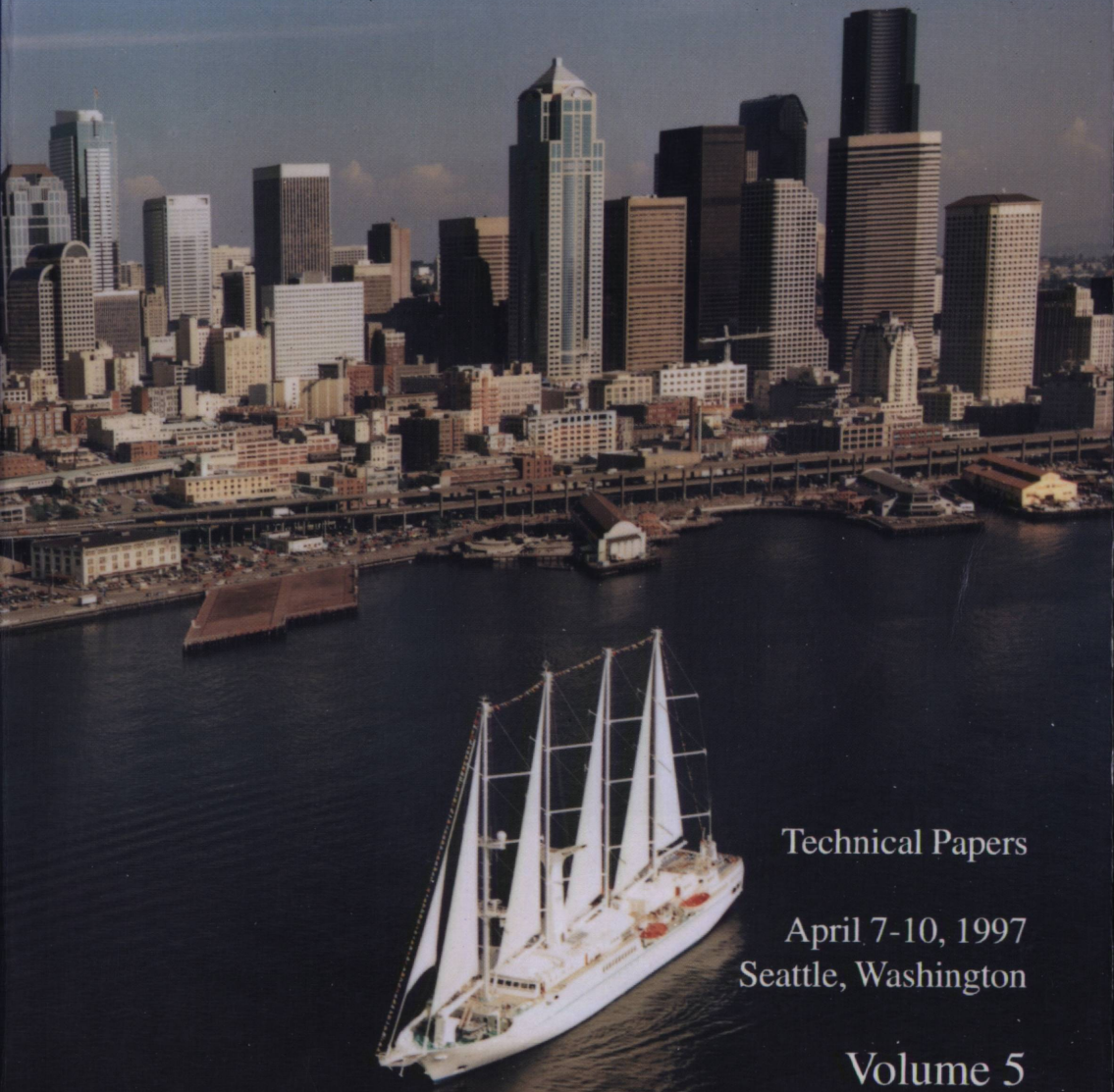
1997 ACSM/ASPRS

Annual Convention & Exposition



Technical Papers

April 7-10, 1997
Seattle, Washington

Volume 5
Auto-Carto 13

Sponsored by:

Resource Technology

1997
ACSM/ASPRS

Annual Convention & Exposition
Technical Papers

ACSM 57th Annual Convention
ASPRS 63rd Annual Convention

Seattle, Washington
April 7-10, 1997

Volume 5
Auto-Carto 13

ISBN-1-57083-043-6 (set)
ISBN-1-57083-048-7 (vol. 5)

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

Printed in the United States of America

Cover image of Seattle, Washington courtesy of Walker and Associates of Seattle, Washington. Walker is part of the AeroMetric Group of companies which include AeroMetric, Inc. of Sheboygan, WI; AeroMetric, Inc. of Fort Collins, CO; Air Survey of Sterling, VA; AeroMap US of Anchorage, AK; and, MARKHURD of Minneapolis, MN.

Table of Contents

Author Index

TOWARD A NETWORK MAP ALGEBRA

Marc P. Armstrong
Department of Geography and
Program in Applied Mathematical
and Computational Sciences
The University of Iowa
Iowa City, IA 52242
marc-armstrong@uiowa.edu

Paul J. Densham
Centre for Advanced Spatial
Analysis
Department of Geography
University College London
26 Bedford Way
London WC1H OAP
pdensham@geog.ucl.ac.uk

ABSTRACT

When decision-making groups work together to solve location-selection problems they often experience difficulty in understanding the elements of alternative solutions to problems that are held in common by different stakeholders. Because location selection problems are often addressed using decision support tools that represent the supply of services, and demand for it, on a street network, we have developed an approach, called network map algebra, that can operate on collections of alternatives in order to synthesize, summarize and differentiate between entire solutions or parts of them. This network map algebra uses abstracted representations of solutions in the form of vectors and matrices and requires a set of transformations between these abstracted structures and the conventional structures used by vector-based GIS software.

INTRODUCTION

Groups of decision-makers often meet to address complex, ill-structured problems. Although individuals working in isolation normally lack the breadth of experience required to solve such problems, when they work as a group they can pool their expert knowledge, explore their unique perspectives and viewpoints on a problem, and work to resolve the conflicts that result from their differing backgrounds, experiences and opinions. Since complex problems of the type that are routinely encountered in the formulation of public policy often have important spatial components, several issues must be resolved before groups can most effectively use existing geographic information handling and analysis tools to support their decision-making efforts. One important problem is that current software systems do not provide the kinds of geographic tools that are required to support group interaction, visualization and consensus-building activities. The purpose of this paper is to develop the structures and operations needed to generate maps that synthesize the results of a set of location selection scenarios created by either an individual, or two or more group members. Because location-selection algorithms often use networks to provide the spatial structure of movement between supply and demand locations, we place a particular emphasis on the development of a conceptual framework for a network map algebra (NMA) which facilitates the production of maps that are designed to assist group decision-making.

BACKGROUND AND DERIVATION

Within the current GIS paradigm, thematic maps are created by assigning symbols to geometric objects that reflect their attributes; for example, a polygon such as a census tract might be shaded to depict average income or a city might be symbolized with a circle proportional to its population. These methods rely on a set of visual variables that are used to guide the choice of appropriate symbols (MacEachren, 1995) and are now well established in GIS and desktop mapping software. In contrast, many methods of locational analysis operate on the attributes of objects and an abstracted form of their topological relations (Densham and Armstrong, 1993). Figure 1 illustrates a multi-step transformation, performed routinely by GIS software, that converts a simple map into a tabular representation that can be used to support analytical NMA operations. In the first step the map is recast into a discrete set of topological nodes. These nodes are uniquely identified and a simple table (ID and locational coordinates) is derived from this representation.

To generate map solutions to locational analyses, topological relations in the form of a list of allocations of demand locations to facilities, and attributes in the form of a tabular structure that specifies the volume of demand that each facility serves, are created during analysis and must be reconciled with the geometrical representations of objects in a geographical database (demand locations, facilities, and the underlying transportation network). This process of reconciliation can be accomplished through a series of transformations between maps, abstract data structures, operations on these structures and transformations back to mapped representations. In this section we describe this transformational view (see also Tobler, 1979; Clarke, 1995).

Networks are constructed from nodes, points, links and chains. These constitute a set of cartographic primitives that can be analyzed independently or assembled into higher-level structures based on results from analytical operations. For example, a shortest path is computed through a network based on topological connections and distances along chains between nodes. The shortest path between two nodes that lie on different sides of a city, therefore, is a kind of compound object that is built from chains, nodes and their respective topological connections. Likewise, a scenario result from a locational model is reported as a tabular set of allocations of demand locations to one or more supply sites. Thus, for any given supply site the links between it and demand locations, when related to geometrical representations stored in a geographic database (e.g. TIGER files) constitute a compound object that describes the particular geographical characteristics of one possible solution to a location-selection problem. Typically, tabular structures that result from analyses are linked to geometrical representations to facilitate the creation of geographical visualizations (GVIS: MacEachren, 1995) that support locational decision-making (Armstrong *et al.*, 1992).

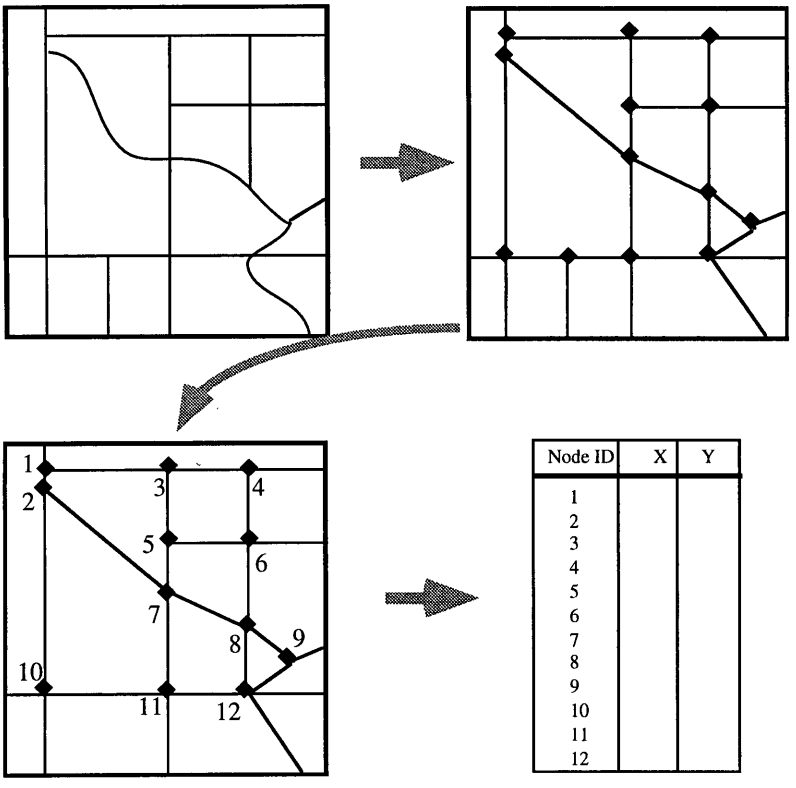


Figure 1. Map abstraction to topological nodes.

A network map algebra must be designed to operate on the structures, vectors and matrices that are derived from the topological relations produced by locational models. In a solution to a locational problem, for example, each supply location (facility) is topologically linked to a set of demand locations. Figure 2 illustrates the process through which data are used to create a simple GVIS or to provide input to locational modeling software, such as LADSS (Densham, 1992). The results of such models are often produced in tabular form and specify the assignment of demand to supply locations while satisfying some user-specified optimization criteria such as minimization of average distance between demand and supply locations. This tabular representation is then used with appropriate geographical information to produce maps that show, for example, the relationship between supply and demand locations.

In Figure 3 a tabular structure that specifies the identity of a set of nodes can be used to generate a crude map as well as raw data that is input to a location model. The model then computes a set of allocations, based on user-specified criteria, of demand to supply locations has been written in tabular form. However, an additional transformation, also shown in Figure 3, and one that is of particular interest in a NMA context, is a simple binary representation of these relationships between supply and demand locations.

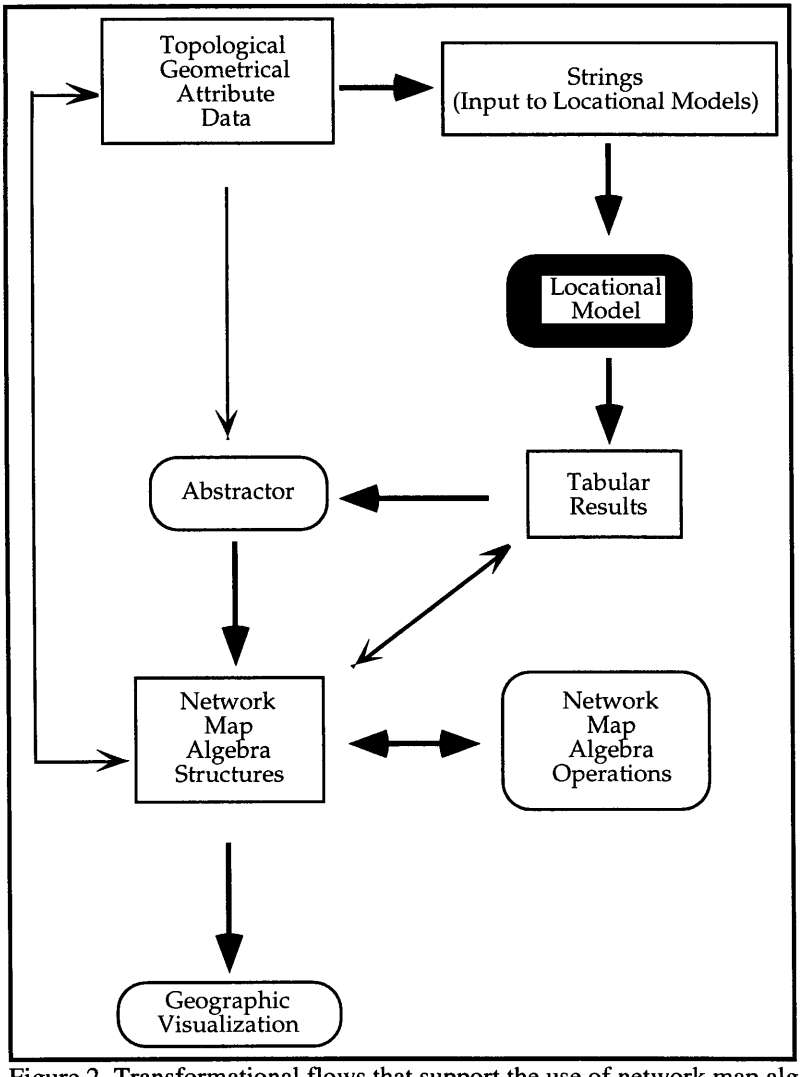


Figure 2. Transformational flows that support the use of network map algebra. Heavy arrows indicate primary flows while lighter arrows indicate flows that are used less frequently. Rectangles indicate data; rounded rectangles indicate processes.

Several types of these transformed representations of solutions are used in NMA operations. Figure 4 shows the derivation of *supply vectors* for a set of three solutions. In this case each node that is specified as a supply site in a solution is denoted with a "1". Blanks are treated as zeros. More complex structures also can be derived to distill relationships between demand and supply locations. In Figure 5 links between supply and demand locations for the same three solutions are represented as binary *allocation of demand* matrices (in this

case a "1" represents the allocation of a demand location to a facility). In this way, several characteristics of each solution can be reduced to a matrix form that is then subjected to further analysis using NMA operations. Other types of analytical structures also can be generated (e.g. demand vectors, network shortest paths and second-order neighbors) with the resulting transformations used with NMA operations to move between alternative representations of a problem and its solutions as required by users as they either visualize or analyze their data.

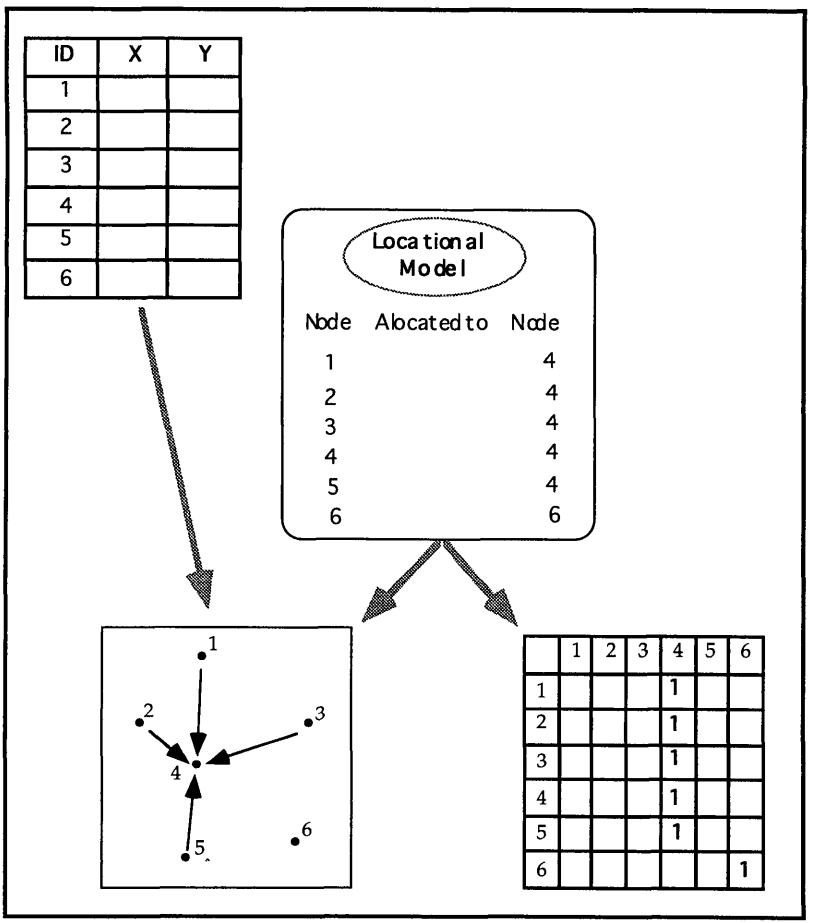


Figure 3. Transformational view of the location-selection process. Solid arrows indicate allocation of demand to a supply location.

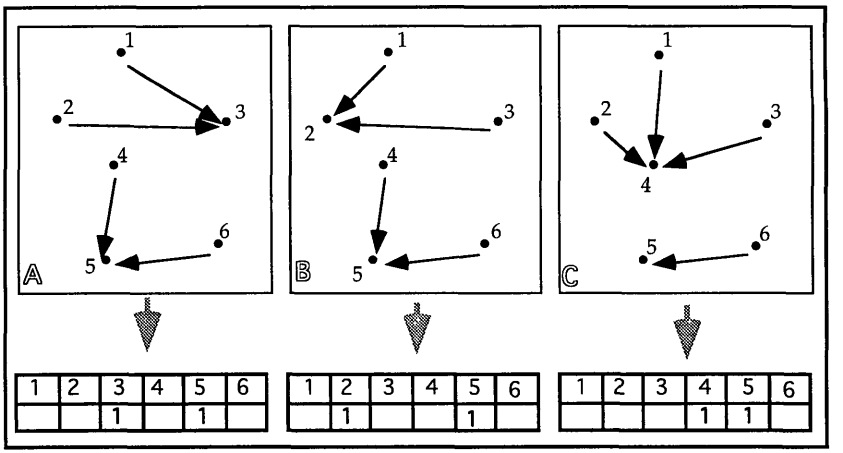


Figure 4. Binary supply vectors

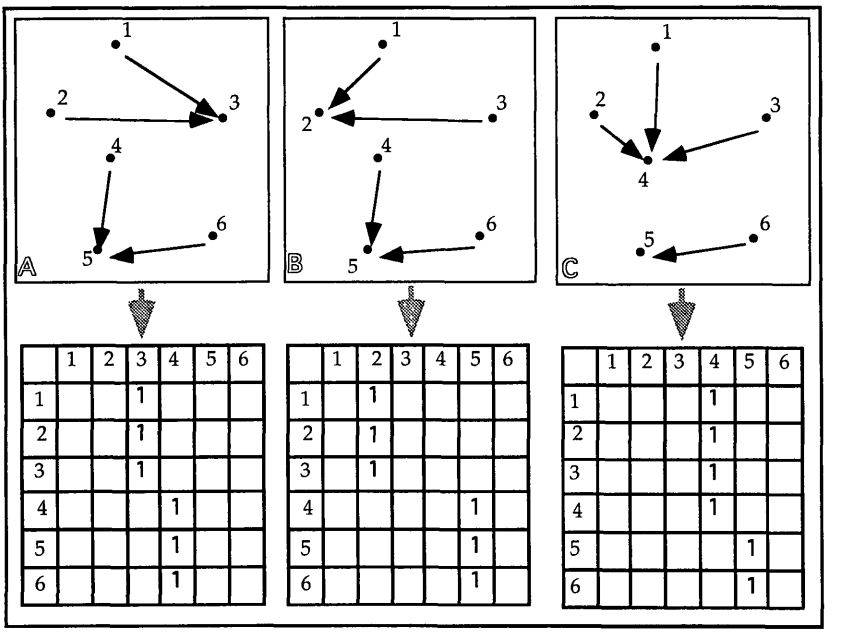


Figure 5. Creating allocation of demand matrices.

ILLUSTRATION

In this paper the primary focus is placed on nodal NMA operations. These are in many ways analogous to Tomlin's focal operations as described in several specifications of map algebra and cartographic modeling that have been

developed for polygon and grid data (see for example, Burrough, 1986; Tomlin, 1990; 1991). In particular we describe how the network map algebra is used to generate two types of maps (facility frequency and allocation consistency) that are designed explicitly to support group decision-making. For example, maps that synthesize the characteristics of a collection of scenarios can be created by applying operations such as "*add* solution_5 *to* solution_7 *to produce* sum_9". The resulting maps enable decision-makers to identify, for example, locations that occur frequently, and are thus robust with respect to a range of objective and subjective criteria, and to highlight those areas over which there is disagreement. This enables group members to work towards the resolution of conflicts and supports the process of consensus building.

Facility Frequency Maps

Facility frequency maps depict those nodes that are selected as facility locations in two or more solutions. In the two-scenario case, a pair of supply vectors is summed, using a local operation on each node, to yield a third vector from which a map is produced. Figure 6 shows how several alternatives, which might be generated by a single user or by different stakeholders in a multiple-participant public policy problem, could be subjected to network map algebra analysis to determine those places in which agreement occurs among the participants about the location of a facility. This summation of supply vectors, yields information about the common elements that exist across the set of solutions and can be linked to a geometrical representation to produce a facility frequency map that is used to support discussion about the solutions generated. Note also that the link between the NMA-generated structures and the data from which they are derived enables the creation of other map types that depict, for example, the volume of demand served at each facility.

Allocation Consistency Maps

Allocation consistency maps depict how often a demand location is allocated to the same facility across a range of scenarios or, conversely, how stable a facility's service area is under a range of allocation criteria. Focal operations are also used to produce these maps. Clearly, in many cases the results will be somewhat chaotic with mapped allocations visually crossing each ether, thus leading to considerable confusion in any map created directly from the raw allocations. However if a measure of saliency is used, the resulting maps will be far more simplified. As shown in Figure 7 a simple threshold operation on the resulting summed allocation of demand matrix reduces the number of allocations that are symbolized, thus leading to a less cluttered appearance.

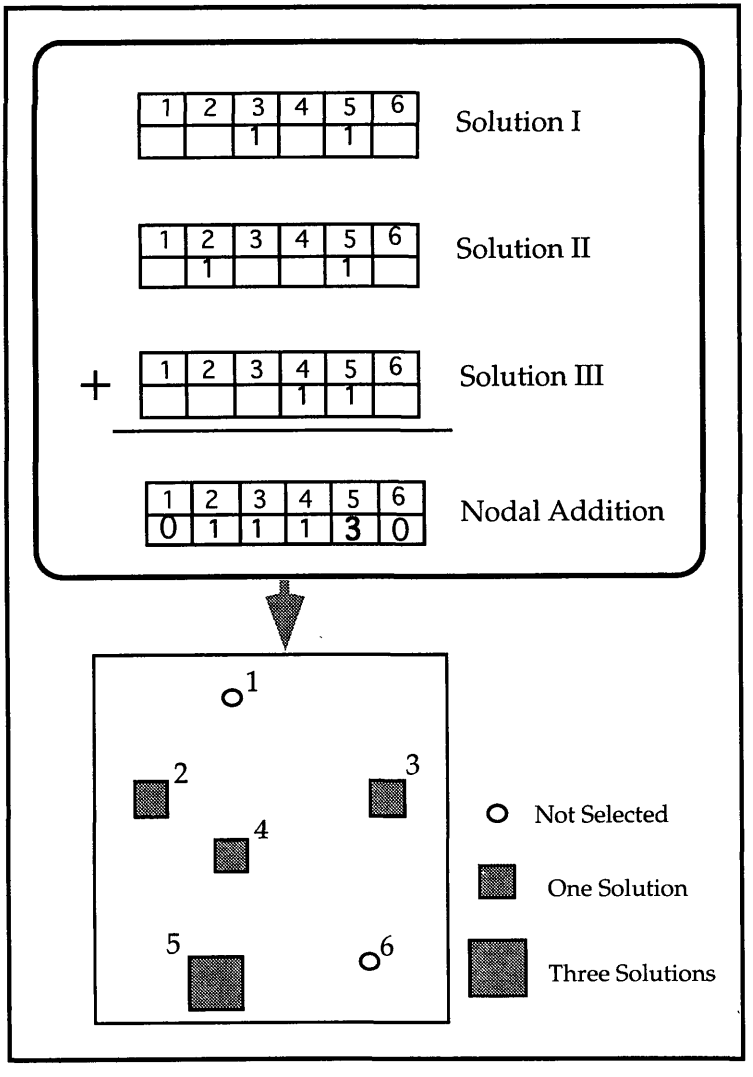


Figure 6. Creation of a facility frequency map using nodal network map algebra operations.

Nodal NMA operations can also be used to produce link-based maps. A network spider map, for instance, depicts maps through a network that link each demand location with its allocated supply location. To build such maps, a matrix-based shortest path algorithm (Arlinghaus *et al.*, 1990) is supplied with origins from the solution's binary supply vector and the requisite destinations from the column in the allocation of demand matrix. When the algorithm identifies the shortest path between facilities and demand locations the relevant links in the network are output and can then be symbolized.

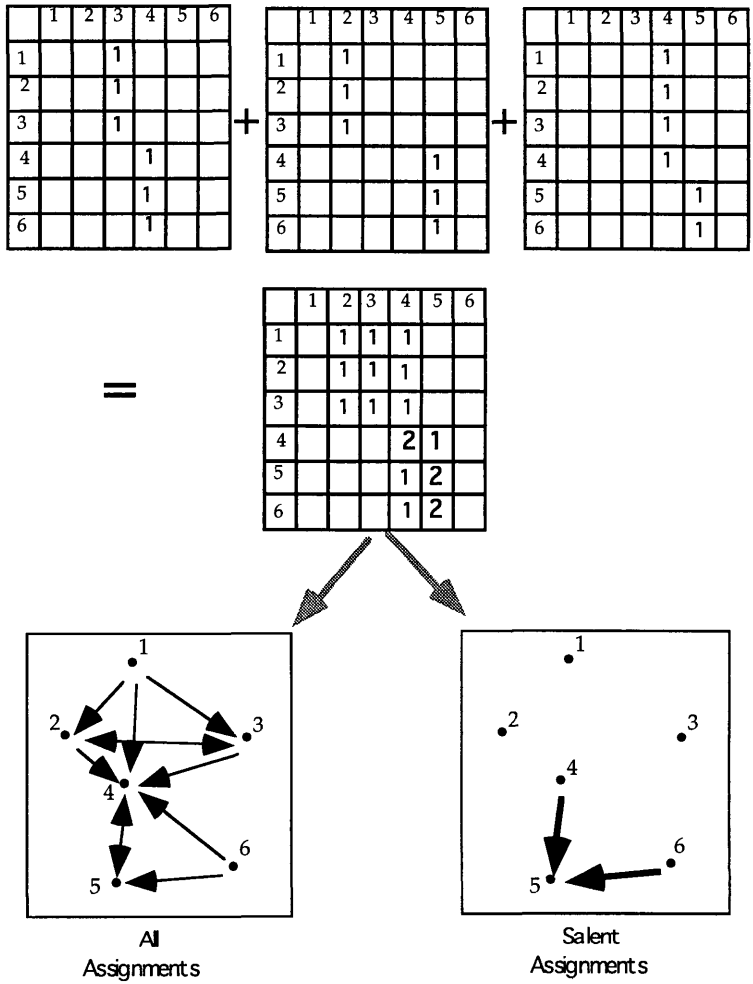


Figure 7. Creation of allocation consistency maps using network map algebra operations.

CONCLUDING DISCUSSION

We have described the rudiments of a network map algebra that supports the process of synthesizing solutions to network-based location-selection problems. This algebra relies on the transformation of scenarios into a set of matrix representations that are then manipulated by operators. These abstracted realizations of solutions to location-selection problems can be passed among users as tokens that are used to generate maps. The entire detailed network (e.g. TIGER file) need not be passed each time a solution is developed. Rather, the network representation is static and the structures are used to symbolize results, thus resulting in a substantial reduction in network traffic.

REFERENCES

Arlinghaus, S.L., Arlinghaus, W.C. and Nystuen, J.D. 1990. The Hedetniemi matrix sum: An algorithm for shortest path and shortest distance. *Geographical Analysis* **22** (4): 351-355.

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

Armstrong, M.P. and Densham, P.J. 1995. Cartographic support for collaborative spatial decision-making. *Proceedings of the 12th International Symposium on Automated Cartography* (Auto-Carto 12), Bethesda, MD: American Congress on Surveying and Mapping, pp. 49-58.

Burrough, P.A. 1986. *Principles of Geographical Information Systems for Land Resources Assessment.* New York, NY: Oxford University Press.

Clarke, K.C. 1995. *Analytical and Computer Cartography* (2nd Edition). Englewood Cliffs, NJ: Prentice Hall.

Densham, P.J. 1992. *The Locational Analysis Decision Support System (LADSS).* NCGIA Software Series S-92-3, National Center for Geographic Information and Analysis, Santa Barbara, CA.

Densham, P.J. and Armstrong, M.P. 1993. Supporting visual interactive locational analysis using multiple abstracted topological structures. *Proceedings of the Eleventh International Symposium on Computer-Assisted Cartography* (Auto-Carto 11), Bethesda, MD: American Congress on Surveying and Mapping, pp. 12-22.

Haggett, P. and R.J. Chorley, 1969. *Network Analysis in Geography* (Edward Arnold, London).

MacEachren, A.M. 1995. *How Maps Work: Representation, Visualization, and Design.* New York, NY: The Guilford Press.

Tobler, W.R. 1979. A transformational view of cartography. *American Cartographer* **6** (2): 101-106.

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

Tomlin, C.D. 1991. Cartographic modelling, in D.J. Maguire, M.F Goodchild and D.W. Rhind, Eds., *Geographical Information Systems: Principles and Applications* (Longman: Harlow), pp. 361-374.

HIGHER ORDER FUNCTIONS NECESSARY FOR SPATIAL THEORY DEVELOPMENT

Andrew U. Frank
Dept. of Geoinformation
Technical University Vienna
frank@geoinfo.tuwien.ac.at

Abstract

The tool we use influences the product. This paper demonstrates that higher order functions are a necessary tool for research in the GIS area, because higher order functions permit to separate the treatment of attribute data from the organisation of processing in data structures. Higher order functions are functions which have functions as arguments. A function to traverse a data structure can thus have as an argument a function to perform specific operations with the attribute data stored. This is crucial in the GIS arena, where complex spatial data structures are necessary. Higher order functions were tacitly assumed for Tomlin's Map Algebra.

The lack of higher order functions in the design stage of GIS and in the implementation is currently most felt for visualization, where the problems of the interaction between the generic computer graphics solutions and the particulars of the application area preclude advanced solutions, which combine the best results from both worlds. Similar problems are to be expected with the use of OpenGIS standardized functionality.

This paper demonstrates the concept of higher order functions in a modern functional programming language with a class based (object-oriented) type concept. It shows how the processing of data elements is completely separated from the processing of the data structure. Code for different implementations of data structures can be freely combined with code for different types of representation of spatial properties in cells. The code fragments in the paper are executable code in the Gofer/Haskell functional programming language.

1 Introduction

The tool used influences the product. This is true for industrial production as well as for research. Nevertheless, there is little discussion in the research community about the tools they use - e.g., formalization methods - and how they influence the research directions. Most research in Spatial Information Theory (Frank and Campari 1993)

is carried out using first order languages. I feel that this restriction is an unnecessary limitation, which makes researchers focus on static relations and precludes adequate treatment of processes. Higher order functions can take this hurdle, but are generally useful to help with the design of GIS architecture. This paper introduces the concept of higher order functions and demonstrates their use for the separation of treatment of data structure and object data. Other beneficial uses of higher order functions are left to be treated in other papers.

Higher order functions are functions which have other functions as arguments. Many programming languages (e.g., Pascal, C, C++) allow such constructions, but generally they are added *ad hoc* and not fully integrated. Higher order functions are well understood mathematically and functional programming languages cleanly implement these concepts.

Research in GIS has been hindered by a mixture of discussion levels: application specific issues, problems of efficient implementation and topics from spatial theory are all treated at once. The connection between these topics is clearly necessary to avoid developing theories for which there is no use in the real world, or the development of tricky solutions without the benefit of a theoretical understanding. But the resulting breadth of the arguments makes it often difficult to detect the essential features.

The formalization in GIS is most often carried out only at the implementation stage. The programmer must resolve all details, from application details to fundamental issues in mathematics. Different concerns are unseparably intertwined: GIS code is complex to understand, loaded with detail, and seldom useful to gain insight. If any formalization is attempted, GIS research has used most often first order languages for formalization (Egenhofer and Herring 1991; Frank 1996a; Pigot 1992; Worboys 1992). It is tacitly assumed that the described functions are integrated in a program system, but the overall architecture of this system cannot be described in a first order language.

Higher order functions are a powerful conceptual tool to separate concerns at different levels, for example data structure and processing of data elements embedded in the data structure. This is crucial for GIS, which are large data collections and require specific, highly optimized and, in consequence, complex data structures (Samet 1989a; Samet 1989b). The design of the operations treating the attributes of the features and the design of the processing of the data structure must be separated and solutions freely combined. For computer cartography, higher order functions open a door for the separation of the different concerns in the map rendering process: graphical issues, management of screen real estate, geometric map projection etc.

This suggests that higher order functions are a necessary tool for systematic research in geographic information systems and their theory. Tomlin in his Map

Algebra (Tomlin 1983a; Tomlin 1983b; Tomlin 1989) has tacitly used higher order functions and one can attribute some of the success of this concept to the clarity of the resulting framework. Map Algebra cannot be formalized without higher order functions (or some rectification of it). Higher order functions are important also to design visualization systems for cartography, where computer graphics tools and cartography demands must be linked. They will be important to combine the generic GIS operations in the forthcoming Open GIS modules.

This paper demonstrates the concept of higher order functions in a modern functional programming language. As an example for demonstration, the separation of operations on pixels and the traversal of the data structure (e.g., quadtree) are explained. It shows how the processing of data elements is completely separated from the processing of the data structure. Code for different implementations of quadtree structures can be freely combined with code for different types of representation of spatial properties in cells. Fragments of actual executable code are given in the Gofer/Haskell language(Hudak et al. 1992; Jones 1991). This is a modern functional programming language, which is class based to provide an object-oriented type concept.

The paper is structured as follows: the next section introduces the concept of higher order functions. Section 3 introduces functional programming languages and gives some examples for higher order functions. The next section introduces the fundamental operations `map` and `fold`, which apply a function to a data structure. Section 5 applies this to operations from the Map Algebra and Section 6 sketches an application to GIS query language and visualization, followed by conclusions.

2 Higher Order Functions

Higher order functions are functions which take functions as arguments. Higher order functions are the discriminating property of higher order languages, which means that first order languages do not permit to pass functions as arguments into functions.

As a metaphor, higher order functions can be seen as mechanical power tools, into which different drills or blades can be inserted to perform different operations. A simple example: a traversal operation is accessing every element of a list and applies an operation which is passed as an argument to every list. This operation can be used to double the value of every element in the list (when a function which multiplies by 2 is passed), can set all values to 0 (when the function returns always the value 0), reduce the values by 1, etc. This is similar to the instruction in everyday life to clean dishes ('take each dish in sequence and wash and dry it'). It is also the operation to draw a map: "Take these (selected) objects and draw each of them according to the scale and legend".

A formal language is called first order, if the symbols can range only over individuals (this precludes quantification over functions and functions which have functions as arguments). It is called second (or higher) order if symbols can range over relations and functions. The ordinary (first) order predicate calculus is a first order language; relational database and Prolog are using first order languages as bases.

The concept of higher order functions is so powerful that even standard programming languages (like C (Stroustrup 1986 p.127), C++ (Ellis and Stroustrup 1990) or Pascal (Jensen and Wirth 1975)) contain it. There are constructs provided which allow to pass a function (or procedure) into another procedure or function as an argument, but the integration of this element of higher order logic with the remainder of the programming language is typically restricted. In Pascal (Jensen and Wirth 1975), the use of this tool is limited: 1) functions cannot be assigned to variables, 2) passing functions circumvents some type checking and is therefore insecure. In C++ a special 'iterator' concept is provided (but tricky to use) to save the programmer the difficulties with passing functions as parameters. The programming languages used for implementation are based on variables and statements and functions remain second class citizens.

3 Functional Programming Languages

The function is the fundamental building block of a Functional Programming Language: everything is a function; functions with arguments, functions without arguments (which are constants) and functions which produce functions as result. Functional programming languages are as old as Fortran. The functional languages, which are strictly typed, use a type system in which functions have proper type and type checking includes the passing of functions as arguments (Milner 1978). Examples of functions are the function `add (+)`, which has two arguments and as a result computes the sum of the two arguments. Its type is written as `(+)::Int -> Int -> Int`. A function can also be user defined, e.g., a function `f (x)` which computes `3x + 2` and has type `f:: Int -> Int`. The constant π is a function with type pi::Float.

To introduce the concept of a higher order function, a function which applies a given other function twice is used. It is demonstrated with the functions to increment `inc`. The code is written in the language Gofer (Jones 1991), which is related to Haskell (Hudak et al. 1992)[1]

```
twice :: (Int -> Int) -> Int -> Int
twice x a = x (x (a))
inc :: Int -> Int
inc x = x + 1
twice inc 4 --->> 6
```

[1]Functional programming languages write functions and their arguments without parentheses: `f x`. Parentheses are only used for grouping expressions and do not indicate function application as in C++.

The fundamental operation is the evaluation of an expression, composed of some functions. An `if-then-else` expression and recursion are the fundamental control structures. Recursive data structures like sequence or tree fit best. The power of functional programming language is often attributed to built-in operations to treat lists; dynamic arrays were predefined with a very powerful set of operations in APL. As an example, a list and a tree are defined recursively (these data types are typically predefined). Code to sum the elements in the list and to count the leaves of a tree are given

```
data List a = Empty | Element a (List a)
data Tree b = Leaf b | Branch (Tree b) (Tree b)
sumList (Empty)= 0
sumList (Element x xs) = x + (sumList xs)
countTree (Leaf b) = 1
countTree (Branch a b) = (countTree a) + (countTree b)
```

4 Higher Order Functions to Map Operations to Data Structures

Complex objects are described by a collection of values, collected in a data structure. Much of the GIS literature is concerned with data structures for geographic data and the efficiency of particular operations on these data structures. The introductory examples here are a polygon as a list of coordinate pairs and a tree with the names and populations of towns in a county.

Operations on a data structure consist typically of code to traverse the data structure, i.e. code which decides which data element is considered next, and code which deals with a single data item. There are two variants, which are often used :

- map: e.g., each coordinate pair of a list is to be scaled.
- fold: e.g., the total population of all the towns in the tree of towns is summed.

Both these operations could be written easily in Pascal as loops over an array, but for realistic applications, more complex data structure will require more code. This code is essentially the same for any operation of this kind for a given data structure. Many lines of Pascal or C++ programs are filled with code controlling the traversal of data structures.

Higher order functions permit to separate the coding of the traversal of the data structure from the operation on the data element. In a functional programming language, the code for these two operations is:

```
scaledCoordList coordList scale = map (scaleTransformation scale)
coordList where
      scaleTransformation scale (Coord x y) = Coord (scale * x)
(scale * y)
totalPop statePop = fold ((+).pop) 0.0 statePop
```

Here, the `CoordList` contains the coordinate pairs to scale, `scaleTransformation` is the operation to change the scale; `statePop` contains the state population by county, `pop` gets the population figure from the record for a

county. The population values are then summed up (starting the count with 0). The two higher order functions `map` and `fold` are defined in the next subsections.

4.1 Map Function

An operation φ is applied to each element in a structure and yields a new data value. The result is a data structure of the same characteristics, but with different values possibly of different type. Examples: a list of reals can be rounded into a list of integers, coordinates in a list can be scaled (as above), or the record structure changed, e.g., to replace records with county name, population and area with records, which contain county name and population density.

The higher order function `map` applies function with signature phi::(a->b) to a data structure `f` with elements of type `a`. The result is a data structure `f` with elements of type `b`[2].

```
map :: (a -> b) -> f a -> f b
```

In many cases φ is a function which returns the same type as its input `(phi:: a->a)` and the result of mapping `a` on the data structure `f a` is again a data structure with type `f a`. This is, for example, the case, if coordinate pairs are scaled and the result is again a coordinate pair.

The `map` function defined above must be specialized for the data structure used. Assuming the definition for list and tree given above, mapping an operation `phi` to the list is simply applying it recursively to each element, mapping the operation `phi` on a tree is applying it to each leaf:

```
map phi (Empty) = Empty
map phi (Element x xs) = Element (phi x) (map phi xs)
map phi (Leaf b) = Leaf (phi b)
map phi (Branch x y) = Branch (map phi x) (map phi y)
```

With these definitions the code for scaling a list of coordinate pairs given above works. It can be used to update the population count of a list of cities with an operation `updatePop`, which is then mapped to the list of Cities[3]:

```
updatePop (City name area pop) = City name area (pop*1.1)
updatePop listOfCities = map updatePop listOfCities
```

4.2 Fold Function

To compute the total population of a county the population of its cities must be added. The input is a data structure, not a data structure as for `map` above. This

---

[2] In order to use higher order functions effectively, the type system must allow parametrized types, i.e. lists of integers, lists of reals etc. In the example f a means a data structure of f with elements of type a.

[3] `updatePop` is used here in a polymorphic fashion: `updatePop` applied to a data element of type city uses the operation definition of the first line, `updatePop` applied to a list of cities, uses the second definition, which calls the first for each city in the list. The type system of a polymorphic language controls this.

operation is called `fold`, because it reminds of folding a piece of paper over and over (Figure 1):

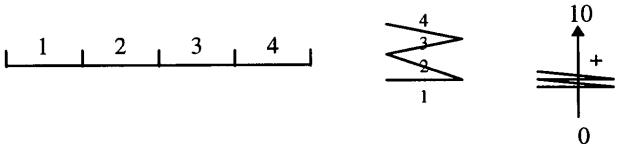


Figure 1: Folding operation

The operation applied in `fold` combines the current value with the result of the previous application. A start value must be given[4] (for the sum above, it must be 0). The standard example is the calculation of the values of the figures from 1 to 100:

```
fold :: (a -> b -> b)-> b -> f a -> b
fold (+) 0 [1..100]
```

The higher order function `fold` has as a first argument the function, as a second argument the start value and as a third argument the data structure. It may be necessary to have two `fold` operations, which operate from right to left and from left to right. The function which is used to fold the data structure must have the signature `psi :: a -> b -> b`, having one argument which is the same type as the data in the data structure and a second argument, which has the same type as the result (these two types can be the same, and most often are). This is necessary to make the result of one application the input for the next application of the function (with the next data value from the data structure).

The definitions for the list as defined above is:

```
fold :: (a -> b -> b) -> b -> f a -> b
fold f z (Empty) = z
fold f z (Element x xs) = fold f (f x z) xs
```

4.3 Combinations of Operations

To compute the total population from a tree of records with county name and population count one can proceed in two steps: 1) map from the record a single population count (with an operation `pop`) and 2) fold with `(+)`.

```
pt = map pop popTree
totPop = fold (+) 0.0 pt
```

Most functional programming languages are referentially transparent. Therefore equals can be substituted with equals. This makes reasoning about programs similar to reasoning in ordinary mathematics and avoids the complications of the temporal reasoning with pre- and post-conditions necessary for commercial programming languages like Pascal or C++. The rule given in the next line can be applied to the

[4] The start value is typically the *unit value* for the operation used, for (+) the value must be 0, for folding with (*), it should be 1 etc. Mathematicians call a group an algebraic structure, consisting of a set of values, an operation and a *unit value 0*, for which the axioms $a + 0 = a$ and $0 + a = a$ hold.

combination of the two functions above (in `totPop`) and yields the simplification `totPop'`.

```
fold f z (map g xs) = fold (f.g) z xs
totPop = fold (+) 0.0 (map pop popTree)
totPop' = fold ((+).pop) 0.0 popTree
```

There is a similar rule to combine multiple mappings:

```
map f (map g x) == map (f.g) x
```

In general, these simplifications are automatically done by the compilers for modern functional languages.

5 Application to GIS: Map Algebra

Map Algebra does not rely on a raster data structure, but is typically conceptualized as operating on a set of arrays of pixels with the same origin and orientation. The local operations in Map Algebra (Tomlin 1989) apply an unary operation (an operation with a single argument) to a single array, by applying it to each cell, or apply a binary operation to two arrays, by applying the operation to corresponding pixels from each array.

The operations in Tomlin's Map Algebra are independent of the data structure and the particulars of the implementation. They can be applied to rasters stored as a full array, run length encoding or as a quadtree. The different storage methods influence performance, but do not change the result. A rewriting of the Map Algebra using a functional language with higher order functions can bring two advantages:

- Potential for optimization: multiple operations can be executed together using the rules for combination given above. The combination of operations in a single pass over the data can greatly speed up performance as the time consuming access to the data is done only once and thus much disk access (or access to the data over the net) is saved.
- Generalization of the operations to work uniformly over raster and vector data in different data structures and to formally analyze the differences between vector and raster operations in the results and the error propagation.

5.1 Separation of Data Structure and Processing of Elements

To demonstrate the separation of treatment of data elements and traversal of data structure, the calculation of the area of a region stored in a quadtree is shown. An example using run length encoding works along the same lines, but cannot be shown due to the space limitations.

Quadtrees (Samet 1989b) are based on the principle of a 4-way branching tree data structure. It is customary to interpret a quadtree structure as a representation of space, in which the leaf nodes are pixels in a square array (Figure 2). Pixels of higher level represent four times the area of the pixel one level lower.

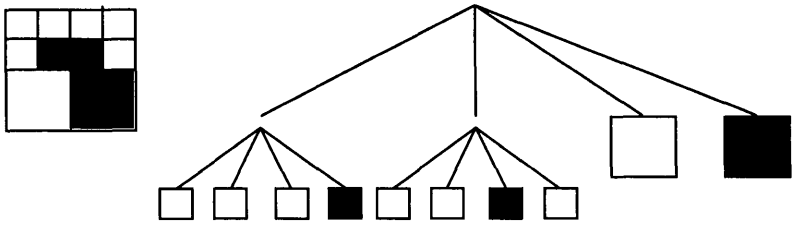


Figure 2 - Example of a quadtree

Recursively, a quadtree is either a leaf (`QL`) or it is a tree with four quadtrees. To keep the code simple, the size of the pixel is included with each leaf of the quadtree; this can be left out in optimized code.

```
data QuadLeaf p = QuadLeaf Int p -- the leaf size and the pixel
value
data Quad p = Q (Quad p)(Quad p)(Quad p)(Quad p) | QL p
```

Code to compute the area uses the corresponding operations for the pixels and corrects for the size of the area.

```
area (Pixel Inside) = 1
area (Pixel Outside) = 0
area (QuadLeaf j p) = j * (area p)

fold f s (QL p) = f p s
fold f s (Q2 q1 q2 q3 q4) = fold f (fold f (fold f (fold f s q1)
q2) q3) q4
area x = fold ((+).area) 0 x     -- x is of type Quad
```

5.2 Operations with two Arguments

The standard `map` function maps a function with a single argument. For the zonal operations of the Map Algebra, functions with two arguments must be mapped. An higher order function `map2` and `fold2` must be defined.

```
map2 :: (a -> b -> c) -> p a -> p b -> p c
map2 f (Empty) x = Empty
map2 f x Empty = Empty
map2 f (List x xs) (Element y ys) = Element (f x y) (map2 f xs ys)
```

With this the corresponding values from two lists can be added or subtracted. Assume you have a list with the population per census block and with the population under 20. The list of the population count (age > 20) is simply

```
map2 (-) totPopulation children
```

The combination of two quadtree or run length encodings is more involved: there are cases when one of the two structures must be expanded to meet the detail level of the other. In order to write this function only once to be used for `map2` and `fold2`, it is factored out in a function `zip`, which takes two data structures as arguments and makes a data structure with pairs of the values found in both structures. `Map2` and `fold2` then use the regular `map` and `fold`, and the function passed as an argument is applied to the two paired values.

```
zip (QL p) (QL q) = QL (p,q)
zip (Q p1 p2 p3 p4) (QL q) = Q (zip p1 q') (zip p2 q') (zip p3 q')
(zip p4 q') where q'= pushdown q
zip (QL p) (Q q1 q2 q3 q4) = Q (zip p' q1) (zip p' q2) (zip p' q3)
(zip p' q4) where p'= pushdown p
zip (Q p1 p2 p3 p4) (Q q1 q2 q3 q4) = Q (zip p1 q1) (zip p2 q2)
(zip p3 q3) (zip p4 q4)
map2 f p q = map f' (zip p q) where f' (a,b) = f a b
fold2 f s p q = fold f' s (zip p q) where f' (a,b) = f a b
```

With these operations, all the focal operations from Map Algebra can be written and applied to arbitrary data structures and feature data types. A choice of functions can be provided in a single program and polymorphism will select the appropriate operations to traverse the data structure and the operation to suit the data type of the feature.

6 Application to Computer Cartography

The examples so far were low level operations in a GIS, very close to the details of the implementation. In this section, I show that higher order functions are also very powerful tools to understand complex, large systems at the highest level of abstraction:

The classical computer graphics program consists of a list of data objects and a visualization loop, which applies the visualization transformation to each object (Foley and Dam 1982; Newman and Sproul 1979) and puts it on the screen. The visualization loop applies a series of transformations to the objects: perspective projection from 3D object coordinates to the 2D coordinates on the screen, clipping of parts which are outside of the viewing area etc. Other transformations apply to the lines or the symbols and produce the desired line style, according to the map legend selected.

These transformations can be written as functions applicable to objects. If the objects to display are in the `listOfFeatures`, the total operation is

```
map display.(scale 50).(clip xmin ymin xmax ymax).(symbolization
symboltable) listOfFeatures
```

This can be expanded to a query language for cartographic application. Simplifying the problem, one can start with a query language which has two query inputs - the selection criteria for the objects to display and the map legend to be used. This operation can be combined from a `filter` operation - a second order function - and `map` (as defined above):

```
filter criteria (Empty) = Empty
filter criteria (Element a as) = if criteria a then Element a
(filter as)                                  else filter as
query legend criteria database = (map (display legend)) . (filter
criteria) db
```

7 Conclusions

This paper introduces higher order functions, i.e. functions which take functions as arguments, in the GIS literature. Such functions are well known in mathematics, but

were not explicitly used for formalization of spatial information theory so far. First order languages are well suited for the analyses of static relations between objects, but they fail when dynamic behavior must be described. GIS must increasingly deal with dynamic objects and higher order functions are therefore necessary, but the same functionality is necessary when describing the behavior of complex, dynamic software systems, like GIS.

As a first example for the importance of higher order functions, this paper demonstrates how higher order functions allow to separate the part of operations specific to the data structure from the code of the operations which is specific to the data type stored. GIS are large data collections and must use complex spatial data structures. It is beneficial to separate the code which traverses the data structure from the code which operates on the feature data.

This is shown using a modern functional programming language and applied to Map Algebra (Tomlin 1989). The examples given are actual code as it can be executed. It shows convincingly the elegance and power of higher order functions. With this tool, the overall architecture of complex systems, e.g., Map Algebra or a cartographic query language, can be described in a single executable system without using *ad hoc* tricks. More examples can be found in (Frank 1996b; Frank 1996c)

The examples given here make clear that it is not sufficient to add a few higher order functions as fixed functions to a programming language, but that the full capability of writing new higher order functions is required. It is further necessary to allow data types with parameters, e.g., *List of Integer* must be differentiated from *List of Char* and a generic *List of x* or even *f of x* (*f* and *x* being type variables) must be possible.

References

Egenhofer, M. J., and J. R. Herring. 1991. High-level spatial data structures for GIS. In *Geographic Information Systems: Principles and Applications*, edited by D. Maguire, D. Rhind and M. Goodchild: Longman Publishing Co.

Ellis, M. A., and B. Stroustrup. 1990. *The Annotated C++ Reference Manual.* Reading, Mass.: Addison-Wesley.

Foley, J. D., and A. van Dam. 1982. *Fundamentals of Interactive Computer Graphics, Systems Programming Series.* Reading MA: Addison-Wesley Publ. Co.

Frank, A. U. 1996a. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal for Geographic Information Systems* 10 (2).

Frank, A. U. 1996b. Hierarchical Spatial Reasoning: Internal Report. Dept. of Geoinformation, Technical University Vienna.

Frank, A. U. 1996c. Using Hierarchical Spatial Data Structures for Hierarchical Spatial Reasoning: Internal Report. Dept. of Geoinformation, Technical University Vienna.

Frank, A. U. , and I. Campari, eds. 1993. *Spatial Information Theory: Theoretical Basis for GIS.* Edited by G. Goos and J. Hartmanis. 1 vols. Vol. 716, *Lecture Notes in Computer Science.* Heidelberg-Berlin: Springer Verlag.

Hudak, P., et al. 1992. Report on the functional programming language Haskell, Version 1.2. *SIGPLAN Notices* 27.

Jensen, K., and N. Wirth. 1975. *PASCAL User Manual and Report.* Second Edition. Berlin-Heidelberg: Springer-Verlag.

Jones, M. P. 1991. An Introduction to Gofer: Yale University.

Milner, R. 1978. A Theory of Type Polymorphism in Programming. *Journal of Computer and System Sciences* 17:348-375.

Newman, W. M., and R. F. Sproul. 1979. *Principles of Interactive Computer Graphics.* New York: McGraw Hill.

Pigot, S. 1992. A Topological Model for a 3D Spatial Information System. In Proceedings of 5th International Symposium on Spatial Data Handling, at Charleston.

Samet, H. 1989a. *Applications of Spatial Data Structures. Computer Graphics, Image Processing and GIS.* Reading, MA: Addison-Wesley Publishing Co.

Samet, H. 1989b. *The Design and Analysis of Spatial Data Structures.* Reading, MA: Addison-Wesley Publishing Co.

Stroustrup, B. 1986. *The C++ Programming Language.* reprinted with corrections July 1987. Reading MA: Addison-Wesley Publishing Co.

Tomlin, C. D. 1983a. Digital Cartographic Modeling Techniques in Environmental Planning. Ph.D. thesis, Yale University.

Tomlin, C. D. 1983b. A Map Algebra. In Proceedings of Harvard Computer Graphics Conference, at Cambridge, Mass.

Tomlin, C. D. 1989. *Geographic Information System and Cartographic Modeling.* New York: Prentice Hall.

Worboys, M. 1992. A Model for Spatio-Temporal Information. In Proceedings of 5th International Symposium on Spatial Data Handling, at Charleston.

INTEGRATING TRADITIONAL SPATIAL MODELS OF THE ENVIRONMENT WITH GIS

Karen K. Kemp
Assistant Director
National Center for Geographic Information and Analysis
University of California, Santa Barbara, USA
email: kemp@ncgia.ucsb.edu

Beginning with the premise that environmental science disciplines have traditionally used conceptual spatial models which are different both from those used in other disciplines and from those provided by the digital data models of current GIS, a consideration of the fundamental role that phenomena which vary continuously across space play in environmental models suggests that continuity may provide an essential unifying theme. This in turn may provide an important basis for the design of interoperable GIS for environmental modeling purposes. Issues arising from a fundament assumption of continuity and themes for continued research are raised.

INTRODUCTION

The increasingly sophisticated tools for spatial modeling and analysis provided by today's GIS are now leading to a revolution in environmental modeling, one which encourages scientists to incorporate spatial processes and relationships in their models. However, the driving force for the design of most widely used GIS packages has not been environmental science. As a result, translation of the unique spatial concepts and models which have evolved independent of GIS in the various environmental sciences is not always obvious or without misapplication. Some effort has been directed recently to the development of software interfaces which will permit translation of generic spatial models such as "field" into the standard data models provided by today's GISs (cf Laurini and Pariente 1996; Vckovski and Bucher 1996). Likewise, the relationship between the real world and how it is represented in GIS has also been the subject of discussion (Burrough and Frank 1995; Couclelis 1992; Csillag 1996; Goodchild 1992; Goodchild 1993; Kemp 1996a; Kemp 1996b; Nyerges 1991; Peuquet 1990). However, these advances have yet to fully address the specific needs of individual environmental scientists as they attempt to make use of spatial information and the new spatial technologies.

This premise behind the research outlined here is that traditional (pre-GIS) approaches used by various environmental modeling disciplines to represent the spatial extents of their phenomena of interest and to implement the interactions between them differ from each other and from those provided by GIS. By examining the underlying bases which led to the development of these discipline specific representations, it should be possible to determine which are the critical aspects needing particular attention in GIS/modeling interfaces. The somewhat surprising conclusion that there may, in fact, be more similarities than differences in how environmental modelers conceptualize space points at a potential basis for true integration of environmental models and GIS.

The following paper documents a preliminary study carried out through in-depth interviews with a range of scientists building or implementing environmental models. The opportunity for this study was provided during a brief sabbatical visit to the Australian National University and CSIRO, both in Canberra, Australia, during October 1996.

PRELIMINARY PREMISES

The following four premises formed the initial basis for the study. A later section explains how these ideas have since been modified as a result of the interviews. The term "conceptual spatial model" refers to the analog models used to constrain or inform data collection activities and/or used during the conceptualization of process models.

1. The conceptual spatial models used by environmental modelers differ in significant ways from the spatial data models provided in current GI systems. Simple mappings between these models are not currently possible. This implies that environmental modelers generally must modify their models in order to use GIS. This is sometimes difficult and may result in incorrect use of available data and misinterpretation of model results.

2. The objects of study, traditional sampling designs and modeling techniques used by individual environmental science disciplines lead to discipline specific conceptual spatial models. These conceptual spatial models vary significantly between disciplines. Thus, there are significant differences in how different sciences discretize space, sample spatially distributed phenomena and extrapolate from their discrete samples to the phenomena being studied.

3. However, it is possible to deconstruct these differences such that the fundamental common characteristics of conceptual spatial models can be identified and measured.

4. These characteristics can be used to develop interoperable interfaces, data models or other elements of GI systems which will enable environmental modelers to use them more efficiently.

In order to find support for these premises and to set the basis for activities related to the fourth item above, the following steps were planned.

1. Describe and characterize conceptual spatial models used by environmental modelers for data collection and for model development
2. Using this information, devise direct mappings between these conceptual spatial models and digital spatial models.
3. Using these mappings, outline some improvements to the design of new interfaces, data models and/or other GIS components in order to improve the efficiency with which environmental modelers can use GIS and to assist in the development of interoperable components for GIS .

CONCLUSIONS ABOUT THE PREMISES

Premise 1: The conceptual spatial models used by environmental modelers differ in significant ways from the spatial data models provided in current GI systems.

By definition, environmental models are environmentally determined. Thus since many environmental phenomena are fields (phenomena for which a value exists at all locations and which may vary continuously across space), environmental models are fundamentally continuous. Hence, environmental *modelers* generally have a continuous view of the world. There are several data models for representing fields in GIS (including cellgrids, planar enforced polygons, TINs, contour lines, pointgrids and irregular points). Methods for manipulating data in these data models are widespread and robust. For example, watershed models which model the flow of water across surfaces are often implemented as finite element solutions (where finite elements are expressed in GIS as polygons). Ground water models likewise often use gridded finite element structures (where finite elements are stored as cellgrids). Therefore, it may be concluded that the discrete digital data models provided by GIS do not present *conceptual* problems to the modelers.

Premise 2: The objects of study, traditional sampling designs and modeling techniques used by individual environmental science disciplines lead to discipline specific conceptual spatial models.

If environmental modelers generally do perceive their phenomena as continuous or see their phenomena as being environmentally determined, then

their conceptual spatial models do not vary significantly between disciplines when analysis depends on a context of the continuous environment. However, the objects of study do vary from superimposed continuously varying phenomena (such as pressure and temperature surfaces in climatology), to objects embedded in continuous matrices (such as faults and intrusions in structural geology), to independent objects (such as individual mammals in wildlife biology). On the other hand, environmental determinism is a fundamental principle in the prediction of the occurrences of many of the phenomena and so they can all be seen to exist within a continuous matrix or at least on a continuous probability surface. Thus, again, continuity provides a common context.

In some sciences, traditional data collection and representation techniques have relied on the discretization of both space and the phenomena being studied. This is particularly true in soil science, geology and vegetation ecology. In these cases, data collection requires experts who interpret the environmental clues, some of them unspecified and unmeasureable, and make conclusions about the distribution of classes of the phenomenon being mapped. The data which is ultimately recorded (i.e. mapped) is not the fundamental observed phenomena, but an inferred classification. An assumption of continuous change across space in the class of the phenomenon does not exist in these data collections.

However, it has long been recognized that this assumption of discontinuity, of homogeneous regions with distinct boundaries, in disciplines such as soils or vegetation science is invalid (cf. Burrough et al. 1977; MacIntosh 1967). These phenomena which are strongly influenced by environmental gradients do vary significantly over space. For many environmental modeling purposes, classified data collection techniques do not result in satisfactory digital records of the phenomena. They do not match the scientists' conceptual models of their phenomena.

Fortunately, the ability to store and manipulate large spatial data bases and the powerful new spatial technologies have begun to allow environmental modelers to move the digital representations closer to these continuous conceptual models. At several different locations, researchers are now working to develop models of soil formation and vegetation growth which are based on continuous environmental determinants such as elevation and rainfall (see for example Burrough et al. 1992; Gessler et al. 1996; Kavouras 1996; Lees 1996; Mackey 1996). These environmental models allow soils or vegetation to be described by a number of different parameters, and, only when necessary, classified accordingly. Classes can be extracted for any set of criteria using various statistical techniques.

If these newest efforts to model soils and vegetation on continuous bases are successful, as seems likely, the contention that significant conceptual differences do not exist between the different sciences themselves becomes even more well founded. However, significant differences do still exist, but these come between the conceptual models of the environmental scientists and those of the environmental managers for whom the models are often developed (Burrough and Frank 1995; Couclelis 1996). At the management end of modeling applications, continuous results are often too difficult to integrate conceptually, particularly when there are several environmental gradients involved. Classification allows many different factors to be summarized and understood conceptually, though not necessarily analytically.

Premise 3: It is possible to deconstruct these differences such that the fundamental common characteristics of conceptual spatial models can be identified and measured.

As the above discussion has asserted, there are no fundamental differences in conceptual models between environmental science disciplines. Thus, environmental phenomena as continuous fields, in some cases with embedded objects, may provide the unifying theme, the fundamental common characteristic.

However, this does not mean everything will need to be represented as continuous fields. It is possible to conceive of and model a continuous environment composed of homogeneous discrete units such as watersheds. It is generally accepted that if the processes being studied operate at a regional scale, subregional size areas below this scale, such as small watersheds, provide sufficient variation for the modeling effort. This, in fact, is the basic premise of the finite element models so widely used in watershed modeling (Vieux et al. 1990). This permits an assumption of homogeneity even within a continuous context and suggests that further consideration should be given to the issue of scale and its relationship to classified continuous phenomena.

Premise 4: These characteristics can be used to develop interoperable interfaces, data models or other elements of GI systems which will enable environmental modelers to use them more efficiently.

Interoperability works best when based on a common conceptual reality. Objects and phenomena should be conceptualized within their physical environment and their attributes and relationships expressed in ways which allow interfaces to translate these generic qualities into system specific values. This means that if reality forms the central interface between different environmental models and spatial databases, all data can be passed through the interface, conceptually returning it to its expression in the physical environment before it is redefined as required for specific software.

Some effort has been directed at itemizing these generic qualities of the physical environment which we seek to model (Burrough and Frank 1995; Couclelis 1996). Methods for quantifying these characteristics are the subject of further research by this author. It is possible to conceive of a software product which would assist environmental scientists and managers to identify and measure the critical characteristics of the environment which determine how it will be modeled and to understand and express the spatial and aspatial components relative to their problems. Such a product might, for example, construct objects (in an OO sense) ready for computation.

ISSUES

These conclusions suggest a number of critical issues which need to be addressed if a functional link between conceptual models and GIS data models can be found.

Is continuity, possibly with embedded objects, "the" conceptual model for environmental modelers?

Do all environmental sciences work in the continuous model? Is geology fundamentally different given that there are discrete geologic objects within continuous matrices as well as continuously varying rock bodies which are discontinuous at boundaries? Can a conceptual temporal model be used to combine and explain intersecting lithologies?

The need for classification remains.

What is the role of classification in sciences with continuous views of their phenomena? Is the need to classify during data collection now unnecessary given current computing power and massive digital storage? What do boundaries mean in the environmental sciences? How do boundaries based on varying criteria affect ecological theory? Does the identification and study of pattern require classification? Must we eventually classify in order to understand?

Classification is a scale issue.

How does scale affect our ability to conceptualize continuous phenomena using discrete representations? Can varying process scales be integrated by using a continuous conceptual model of the phenomena?

Do managers need different spatial models?

Is there a difference between modeling for prediction versus modeling for description and/or management? Do managers need a more discrete (i.e. classified) view of space or do we simply need to educate managers to work with data in forms other than classified maps?

Expert knowledge plays a major role in the understanding and modeling of environmental systems.

How is expert knowledge incorporated into models of processes? What role does it play in classification and data collection? How can we be explicit about the incorporation of expert knowledge in data collection and modeling activities? Can modelers replace or simulate the expert knowledge of the field scientists?

Conceptual temporal models also need to be addressed.

Which sciences assume change and which are static? Historical and episodic events affect the environment but these cannot be represented or modeled well. This is also a scale issue. What about continuity in time? Can space be substituted for time or vice versa (e.g. succession demonstrated by going up an elevation gradient or astrophysical location equating with time)?

Can models be usefully classified as either spatial or aspatial?

Is there a significant difference? Are aspatial models just spatial models at regional scales in which spatial heterogeneity is not relevant at large process scales? How do aspatial models and aspatial data incorporate space? How is space despatialized for aspatial modeling and data collection? How do aspatial models represent changes which have an impact over space?

CONCLUSION

Powerful new tools and paradigm changes are leading to a revolution in environmental modeling. The opportunity to build models which represent continuous variation across the landscape are changing the ways in which we gather data and describe the environment. These in turn provide greater opportunity to provide case-specific information for environmental managers. Much work remains to be done. The results of this preliminary study will inform further detailed research into conceptual spatial models and continuity as an integrating medium. Further work on this theme will be incorporated into efforts related to NCGIA's new research initiative on Interoperating GISs.

ACKNOWLEDGMENTS

These conclusions arise from discussions with the scientists and modelers in Canberra Australia at the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Divisions of Water Resources, Soils, Forestry, Wildlife and Ecology and Information Technology, also at the Australian National University (ANU) at the Centre for Resource and Environmental Studies (CRES) and the Department of Geography. My thanks to everyone who gave me some of their time and many of their ideas. I hope I will have a chance to continue these discussions in the future. Also, I would like to thank CSIRO Division of Information Technology, CSIRO Division of Water Resources, the ANU Department of Geography and the NCGIA who provided support for this research. Research at the NCGIA is supported by grants from the National Science Foundation (SBR 88-10917).

REFERENCES

Burrough, P. A., Brown, L., and Morris, E. C. (1977). Variations in vegetation and soil pattern across the Hawkesbury Sandstone plateau from Barren Grounds to Fitzroy Falls, New South Wales. *Australian Journal of Ecology*, 2:137-59.

Burrough, P. A., and Frank, A. U. (1995). Concepts and paradigms in spatial information: Are current geographical information systems truly generic? *International Journal of Geographical Information Systems*, 9(2):101-116.

Burrough, P. A., MacMillan, R. A., and vanDeursen, W. (1992). Fuzzy classification methods for determining land suitability from soil profile observations and topography. *Journal of Soil Science*, 43(2):193-210.

Couclelis, H. (1992). People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, A. U. Frank, I. Campari, and U. Formentini, eds., Springer-Verlag, pp. 65-77.

Couclelis, H. (1996). Towards an Operational Typology of Geographic Entities with Ill-defined Boundaries. In *Geographic Objects with Indeterminate Boundaries*, P. A. Burrough and A. U. Frank, eds., Taylor &Francis, pp. 45-55.

Csillag, F. (1996). Variations on hierarchies: Toward linking and integrating structures. In *GIS and Environmental Modeling: Progress and Research Issues*, M. F. Goodchild, L. T. Stayaert, B. O. Parks, C.

Johnston, D. Maidment, M. Crane, and S. Glendinning, eds., GIS World Books, Fort Collins, CO, pp. 433-437.

Gessler, P., McKenzie, N., and Hutchinson, M. (1996). Progress in Soil-landscape Modelling and Spatial Prediction of Soil Attributes for Environmental Models. In *Proceedings of Third International Conference/Workshop on Integrating GIS and Environmental Modeling*, Santa Fe, NM. National Center for Geographic Information and Analysis, University of California, Santa Barbara, CA. CD-ROM and WWW.

Goodchild, M. F. (1992). Geographical data modeling. *Computers and Geosciences*, 18(4):401-408.

Goodchild, M. F. (1993). Data models and data quality: Problems and prospects. In *Environmental Modeling with GIS*, M. F. Goodchild, B. O. Parks, and L. T. Steyaert, eds., Oxford University Press, New York, pp. 94-103.

Kavouras, M. (1996). Geoscience Modelling: From Continuous Fields to Entities. In *Geographic Objects with Indeterminate Boundaries*, P. A. Burrough and A. U. Frank, eds., Taylor &Francis, pp. 313-323.

Kemp, K. K. (1996a). Fields as a framework for integrating GIS and environmental process models. Part one: Representing spatial continuity. *Transactions in GIS*, 1(3):in press.

Kemp, K. K. (1996b). Fields as a framework for integrating GIS and environmental process models. Part two: Specifying field variables. *Transactions in GIS*, 1(3):in press.

Laurini, R., and Pariente, D. (1996). Towards a Field-oriented Language: First Specifications. In *Geographic Objects with Indeterminate Boundaries*, P. A. Burrough and A. U. Frank, eds., Taylor & Francis, pp. 225-235.

Lees, B. (1996). Improving the spatial extension of point data by changing the data model. In *Proceedings of Third International Conference/ Workshop on Integrating GIS and Environmental Modeling*, Santa Fe, NM. National Center for Geographic Information and Analysis, University of California, Santa Barbara. CD-ROM and WWW.

MacIntosh, R. P. (1967). The continuum concept of vegetation. *Botanical Review*, 33:130-187.

Mackey, B. (1996). The role of GIS and environmental modelling in the conservation of biodiversity. In *Proceedings of Third International Conference/Workshop on Integrating GIS and Environmental Modeling*, Santa Fe, NM. National Center for Geographic Information and Analysis, University of California, Santa Barbara, CA. CD-ROM and WWW.

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

Peuquet, D. J. (1990). A conceptual framework and comparison of spatial data models. In *Introductory Readings in Geographic Information Systems*, D. J. Peuquet and D. F. Marble, eds., Taylor & Francis, London and Bristol, PA, pp. 250-285.

Vckovski, A., and Bucher, F. (1996). Virtual Data Sets - Smart Data for Environmental Applications. In *Proceedings of Third International Conference/Workshop on Integrating GIS and Environmental Modeling*, Santa Fe, NM. National Center for Geographic Information and Analysis, University of California, Santa Barbara. CD-ROM and WWW.

Vieux, B. E., Bralts, V. F., Segerlind, L. J., and Wallace, R. B. (1990). Finite Element Watershed Modeling - one-dimensional elements. *Journal of Water Resources Planning and Management - ASCE*, 116(6):803-819.

UNDERSTANDING TRANSFORMATIONS OF GEOGRAPHIC INFORMATION

Nicholas Chrisman
chrisman@u.washington.edu
Department of Geography, University of Washington
Seattle WA 98195-3550

ABSTRACT

Transformations have been presented as an organizing principle of analytical cartography. To date, the theories have focused on geometric distinctions, such as point, line and area. This paper presents a new scheme for geographic transformations based on measurement frameworks as the principal distinction. Transformations between measurement frameworks can be viewed in terms of a neighborhood and a rule to process attribute information. This scheme provides a way to organize most of the operations performed by GIS software.

BACKGROUND: TRANSFORMATIONS

While the dominant school of cartography views cartography as a communication process, there has always been another group focused on transformations. The most classic transformation involves the mathematical conundrum of transferring the nearly spherical Earth onto a flat piece of paper, the process of map projection. No cartographic education is complete without a thorough understanding of projections. For centuries, a cartographer could ensure a place in posterity by inventing another solution to the map projection problem. Even Arthur Robinson, whose career was dedicated to thematic cartography and map communication, may have greater recognition for his compromise world projection through its adoption by the National Geographic Society.

The key importance of a map projection is not in the mathematical details. Projections demonstrate how measurements taken on one kind of geometric model can be transferred to another model, subject to certain constraints. For instance, in moving from the earth to a plane, it is possible to preserve either the geometric relationships of angles (conformality) or of area (equivalence), but not both. This operation on geographic data became the basis for Tolber's view of analytical cartography.

While much of Tobler's work dealt with projections, he also advanced a 'transformational view of cartography' (Tobler 1979b) that considered all operations as transformations of information content. Analytical cartography has remained a minority component of the discipline, though some continue to extend it (Nyerges 1991; Clarke 1995).

Analytical cartography developed in the era of Chomsky's transformational grammars, an attempt to systematize linguistics that had far-reaching influence throughout the social sciences. Tobler (1976) set out a systems of transformations based largely upon the geometric component of geographic information. This approach informed the three by three (point, line,area) or four by four (with the addition of volume) matrix in Clarke (1995, Figure 11.1, page 184) and Unwin, among others [Figure 1].

From \ To	**Point**	**Line**	**Area**
Point	Point -> Point	Point -> Line	Point -> Area
Line	Line -> Point	Line -> Line	Line -> Area
Area	Area -> Point	Area -> Line	Area -> Area

Figure 1: Cartographic transformations as viewed by Clarke and Unwin

In this matrix, a buffer around a road would be considered a line-to-area transformation, but so would the conversion from a contour line to a TIN. There is little in common between these operations because the relationships implied by the lines are so different. There is no denying that the geometric primitives are important, but they may not tell all the story. The geometric form of input and output are a weak guide to the actual operation that might be performed. Many of the most complex operations are lumped into the diagonal, along with operations that make very minimal changes. This matrix based on the dimensionality of the objects is clearly insufficient to explain the operations performed in a GIS.

Measurement frameworks

What is missing from the standard explanation is an explanation of the different reasons for using a line to represent geographic phenomena. While there are many reasons, they can be organized in according to the fundamental choices made in obtaining the underlying measurements. Geographic information includes a spatial component, a temporal component, and some set of attributes. As Sinton (1978) recognized, each data model imposes a different set of rules on these components. A measurement framework (Chrisman, 1995, 1997) is a conceptual scheme that establishes rules for control of other components of a phenomenon that permit the measurement of one component.

The broad groupings of measurement frameworks listed in Figure 2 provide a

clearer foundation for transformations of geographic information. The different forms of control have only been recognized in specifics, not as a part of a scheme that is broadly applicable. The role of control is critical to understanding transformations. It is particularly important to note that the representation used my not be the same as the measurement framework. It is quite possible to represent a choropleth measurement in a raster data structure, or a set of pixels as vectors. In both cases, additional losses of resolution and accuracy can occur.

Figure 2: General groupings of measurement frameworks

Attribute Controlled Frameworks	
Isolated Objects	
Spatial Object	Single category distinguished from void
Isoline	Regular slices of continuous variable
Connected Objects	
Network	Spatial objects connect to each other, form topology (one category or more)
Categorical Coverage	Network formed by exhaustive classification (multiple categories, forming an exhaustive set)
Space Controlled Frameworks	
Point-based Control	
Center point	Systematic sampling in regular grid
Systematic unaligned	Random point chosen within cell
Area-based Control	
Extreme value	Maximum (or minimum) of values in cell
Total	Sum of quantities (eg. reflected light) in cell
Predominant type	Most common category in cell
Presence / absence	Binary result for single category
Percent cover	Amount of cell covered by single category
Precedence of types	Highest ranking category present in cell
Temporal Frameworks	
Snapshots	[any other measurement framework can be repeated over time]
Transactions	Discrete events are located freely in time
Relationship Controlled Frameworks	
Measurement by pair	Control by pairs of objects
Triangular Irregular Network	Control by uniform slope (gradient & aspect)
Composite Frameworks	
Choropleth	Control by categories (names of zones) then control by space

TAXONOMIES OF GIS OPERATIONS

The GIS literature has a series of alternative schemes used to present the different kinds of operations. Perhaps the most widely cited is Tomlin's (1983; 1990) *Map Algebra.* This scheme is essentially a sequence to present map operations, ranging from the simple to the complex. The simple operations work on a single map, followed by those that work locally on two maps, and so on. However, Tomlin's scheme fails to include all possibilities (and thus provide the 'algebra' promised), because it forces all measurements into a single raster representation and does not distinguish between a representation scheme and a measurement framework. Furthermore, Tomlin's terminology for the operations becomes a bit obscure for the more complex operations. Goodchild (1987) followed the flow of Tomlin's logic, adding some neglected elements, such as information attached to pairs of objects. Burrough (1992) argued for "intelligent GIS" essentially by recognizing more spatial relationships. Recently, Albrecht has described a method to develop commonalities between GIS operations using a sematic network. This approach seems to rely upon a survey of users, thus is vulnerable to the limited perspectives and training in those surveyed. It still seems worthwhile to develop a taxonomy of GIS operations based on transformations between measurement frameworks.

A Theory of Transformations

Any data model consists of a set of objects, relationships between them and a set of axioms (integrity constraints) that control the meaning of the data. Given data within a particular measurement framework, it is most direct to produce a result in the same framework. Thus, a grid of values with a 10 meter spacing can be most easily processed into another grid with ten meter spacing. To generate a different result, new sets of assumptions may be required. These assumptions are required to fill in the gaps in either space, time, or attributes in the original source.

The theory present here contends that transformations between most forms of geographic information can be performed with two sets of assumptions: one to handle space, thus creating a neighborhood, and the other to handle attributes, a rule of combination. Temporal transformations can be handled as special forms of neighborhoods. Neighborhoods can be defined rather flexibly, following the general scheme of Tomlin - moving from the purely local relationships inside one object through immediate neighbors to more complex relationships based on distance and perhaps other considerations. The rules of combination have not been considered as carefully in the GIS literature. Hopkins (1977) described some of the tools to handle map overlay based on Stevens' levels of measurement, but this scheme does not cover all cases. Rules of combination can be grouped into three broad classes based on the amount of information used in the process (Chrisman, 1997). A *dominance* rule simply selects one of the available values based on some criteria (such as taking the largest value). A *contributory* rule uses all the values, giving each an opportunity to contribute to the composite result.

Addition is the most classic contributory rule. Finally, an *interaction* rule uses not just each value, but the pairwise combinations of values.

This taxonomy of attribute rules serves to explain the differences among the approaches to area-based spatial control frameworks. Once the grid cell is imposed on the landscape, there is some kind of rule that takes all the possible attribute values and picks the value. In some cases, this is a rule like "highest value" (as on an aeronautical chart), which is a dominance rule. In other cases, an optical system adds the energy detected. Thus, the rules are a part of the original geographic measurement as well.

This approach to transformations will be introduced by an example. While a three-by-three or four-by-four matrix can be quickly comprehended, a seventeen-by-seventeen matrix (for all the frameworks listed in Figure 2) is difficult to describe or communicate. A subset of measurement frameworks used for surfaces will illustrate the approach.

The rows and columns of Figure 3 list some of the major alternatives for the representation of surfaces. The first "Points with Z" refers to "Spatial objects" where a continuous surface value is measured at an isolated point feature. The second representation is isolines, closed contours that measure the location of a given surface value. Digital Elevation Matrix (DEM) refers to a regular, spatially controlled measurement of elevations. The fourth is the Triangulated Irregular Network (TIN) whose triangles establish relationships of slope between spot heights.

Figure 3: Surface-oriented transformations

In \ Out	Points (w.Z)	Isoline	DEM	TIN
Points (w.Z)	Interpolation	Interp. & trace	Interpolation	Triangulation
Isoline	Interpolation	Interp. & trace	Interpolation	Triangulation *
DEM	Interpolation	Interp. & trace	Resampling	Triangulation *
TIN	Extraction	Tracing	Extraction	Simplify/ Refine

* denotes a triangulation operation that may produce overly dense triangles without some filtering.

The cells in this four-by-four matrix give a label for the procedure that converts information in the row dimension to the column dimension. The

three-by-three matrix in the upper left (lightly grey) is filled with one form or another of interpolation. This operation provides a good example of how a transformation combines relationships and assumptions (axioms) to produce new information.

Interpolation

Interpolation involves a transformation to determine the value of a continuous attribute at some location intermediate between known points. Part of this process requires relationships – knowing which points are the appropriate neighbors. The other part involves axioms – assumptions about the behavior of the surface between measured locations. The balance between these two can vary. Some methods impose a global model, such a fitting a *trend surface* to all the points. Most methods work more locally. The top left cell in the matrix poses the classical problem: given a set of point measurements, assign values to another set of points. This requires two steps. First one must discover the set of neighboring points for each desired location, using a variety of geometric procedures. Then one must apply some rule to determine the result.

Once the neighbors are collected, the problem of assigning a value resolves itself into the rules of combination. A dominance rule will not yield a smooth surface, since it will assign the same value to a neighborhood (usually the Voronoi polygon). A contributory rule usually involves a distance weighted average of the neighbors. Various forms of interaction rules are in use as well. SYMAP had a much-copied interpolation system that weighted points so that distance and orientation to other points were considered. Each method operates by using certain relationships, plus some assumptions about the distribution of values between points. The differences between various forms of interpolation reflect various assumptions about the nature of the attributes.

The process of producing a DEM with uniformly spaced points is just a special case of interpolation for scattered points. To produce isolines, instead of requesting a value at some arbitrary point, the contour specifies the height, and the interpolation discovers the location. Functionally, this is not very different, since the procedure for a weighted average can be algebraically restructured to give a coordinate where the surface has a given value. The manual procedures for contour drawing involved linear interpolation on what amounts to a triangulation (Raisz 1948). In addition to the interpolation, the construction of isolines requires *tracing*, the process of following the contour from neighborhood to neighborhood. Usually, this procedure involves some assumptions about the smoothness of the surface, since the shape of the contour cannot be really estimated from the original point measurements. Tracing also involves relationships between adjacent contours, even those not created with the same neighborhood of points. Parallel contours imply slope gradient and aspect properties, along with other interactions caused by ridges and courselines (Mark 1986). Thus, tracing contours involves many more relationships than a simple decision about the value at a point.

If the input consists of a set of contour lines, the procedure for scattered points still applies. Interpolation will need to establish neighbors, but neighbors between adjacent contours as well as along the lines. Finding the nearest point on the two adjacent contours does not ensure a correct reading of features such as ridges or courselines. This straight line is a simplification for the line of steepest descent. Linear interpolation then proportions the value between the two contour values.

When the input values are organized in a grid structure, the matrix provides the means to access neighbors directly. To produce output for scattered points, the rules can be applied on the immediate neighbors in the grid. To trace contours, the grid values are used to estimate values in the area between them.

Producing a matrix output from a matrix input is a common requirement. Unlike the vector method where the coordinates can be transformed fairly directly, a matrix is delineated orthogonal to a given spatial reference system and with a given spacing. If a different cell size or orientation is needed, the values will have to be converted by *resampling*. For continuous variables, there is no real difference between resampling and interpolation. Sometimes, a simple dominance rule is used; each new grid cell gets the value of the nearest input grid cell. As long as the spacing is not wildly mismatched, this may produce a reasonable representation. For remotely sensed sources, the 'nearest neighbor' interpolation retains a combination of spectral values actually measured by the sensor. It does mean that each value has been shifted from the position at which it was measured by as much as 0.707 times the original pixel distance. Alternatively, it is common to use a contributory method to weight the change over distance using a various formulae, such as bilinear, cubic convolution, or higher order polynomials. Each function imposes different assumptions about the continuity of the surface.

By contrast with the nature of interpolation problems, a TIN provides its own definition of the neighborhood relationships; it also defines without ambiguity the linear interpolation over the face of the triangle. A transformation from a TIN source has much less work to perform. Once a point can be located inside the proper triangle, it is a matter of extraction. Conversion from one TIN to another is a generalization problem of refining or simplifying the representation inside a set of constraints.

Generalizing from the example of surfaces

As the explanation of surface transformation shows, a transformation can be explained in terms of a neighborhood relationship and a rule to process attributes. Temporal relationships can also be included as a form of neighborhood. This leads to a four-way taxonomy of transformations based on the degree to which the information is inherent in the data model or must be constructed through other kinds of information. This can be seen as a two-by-two matrix based on whether the neighborhood is implicit or discovered and

the attribute assumptions are implicit or external.

Case 0: Transformation by extraction – When the source contains all the information required, it provides both the neighborhood relationship and the attribute assumptions to make a transformation look easy. Extraction is usually unidirectional. For example it is possible to create isolated objects from a topological vector database without much trouble, as long as the desired features are identified somewhere as attribute values.

Case 1A: Transformation based on attribute assumptions – In some cases, the transformation keeps the geometric entities intact, and works just with the attributes of those objects. Some of the steps performed on a base layer of polygons fall into this class, but the simplest form involves a raster with a uniform set of pixels. A common example is the transformation which takes a few axes of continuous spectral data and produces classes.

Case 1N: Transformation with geometric processing only – It is even rarer to use just the geometric component. Given two coverages of polygons, it is possible to convert the areas of one into attributes of the other. This is performed entirely as a geometric procedure, using the identifiers of the polygons to tabulate the areas in the correct attribute columns.

Case 2: Complete transformation – The most interesting forms of transformations are ones that combine geometric (neighborhood) constructions along with attribute rules. The interpolation problem discussed above is an archetype, because the two phases are quite distinct. Areal interpolation also falls into this class, even though it deals with areas not points, the two phases combine in much the same way. This relationship shows that this taxonomy combines those functions which are similar in their purpose, not just in the geometric form of their input.

A buffer around a road is also a complete transformation. It uses a simple neighborhood rule (all space within a certain distance of the road), and a simple dominance rule (areas near any road overrule anything else). A polygon overlay produces the geometric raw material for a suitability analysis. The next steps must take up the combination of the attributes now placed in contact. Various approaches to suitability use dominance, contributory or interaction rules, depending on the fit to the purpose. The general scheme of attribute rules that apply to spatial neighborhoods also apply to overlay processing and simple operations inside one measurement framework. This scheme incorporates Tomlin's successive broadening of neighborhood, but adds the formalization of the attribute rules. The important distinctions are not those of geometric form, but related to the basic structure of how the information was constructed.

CONCLUSION

A unifying scheme for transformations requires only two elements: a geometric neighborhood plus a rule to combine or process attributes. The

rules fall into three classes (dominance, contributory, and interaction) based on the treatment of multiple attribute values. Viewed in this way, the operations of a GIS (including map overlay analysis, neighborhood operations, plus the items now treated as transformations) can all be relocated as various kinds of transformations.

ACKNOWLEDGEMENTS

Parts of this paper first appeared in *Exploring Geographic Information Systems,*© 1997 John Wiley and Sons and are used here with permission.

REFERENCES

Albrecht, J. (1995). Semantic net of universal elementary GIS functions. *Proceedings AUTO-CARTO 12,* 235-244.

Burrough, P. A. (1992). Development of intelligent geographical information systems. *International Journal of Geographical Information Systems 6*(1): 1-11.

Chrisman, N. (1995). Beyond Stevens: A revised approach to measurement of geographic information. *Proceedings AUTO-CARTO 12,* 271-280.

Chrisman, N. (1997). *Exploring Geographic Information Systems.* John Wiley, New York.

Clarke, K. (1995). *Analytical and Computer Cartography.* Second edition. Prentice Hall, Englewood Cliffs NJ.

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

Hopkins, L. (1977). Methods of generating land suitability maps. A comparative evaluation. *American Institute of Planners Journal 43:* 386-400.

Mark, D. (1984). Automated detection of drainage networks from digital elevation models. *Cartographica 21*(3) 168-178.

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

Raisz, E. (1948). *General Cartography.* McGraw Hill, New York.

Sinton, D. (1978). The inherent structure of information as a constraint to analysis. Mapped thematic data as a case study. In *Harvard Papers on Geographic Information Systems,* vol. 6, ed. G. Dutton. Addison Wesley, Reading MA.

Tobler, W. (1976). Analytical cartography. *The American Cartographer 3*: 21-31.

Tobler, W. (1979a). Cellular geography. In *Philosophy in Geography*, eds. S. Gale and G. Olsson, p. 379-386. Reidel, Dordrecht NL.

Tobler, W. (1979b). A transformational view of cartography. *The American Cartographer 6*: 101-106.

Tomlin, C. D. (1983). Digital Cartographic Modeling Techniques in Environmental Planning. unpublished Ph.D., Yale University.

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

THE AUGMENTED SCENE: INTEGRATING THE MAP AND THE ENVIRONMENT

James E. Mower
Department of Geography and Planning and
Geographic Information System and Remote Sensing Laboratory
University at Albany, State University of New York

ABSTRACT

This paper develops a theoretical and technological foundation for a new kind of interactive map, the augmented scene, that overlays acquired imagery of the environment on a perspective model of the surface in real time. The ground position and viewing direction of users are computed automatically, freeing them from the map orientation process. We define augmented scenes, propose some applications for them, discuss their construction and use, determine the effects of positional and directional error on system performance, explore their data structure, and review a prototype system under development.

INTRODUCTION

In the field, map interpretation depends upon the ability of the user to orient a sense-acquired view of the environment with its mapped representation. By establishing correlations between visible objects in the scene and their abstractions on the map, the user registers the two views and builds a cognitive projection between them, reconciling the viewing parameters of the map with the perspective view of the scene. Unfortunately, many potential map users have no training in map orientation and interpretation. Even the attempts of skilled users are sometimes frustrated by suboptimal viewing conditions.

This paper will develop a theoretical and technological framework for a new type of map, *the augmented scene*, that projects cartographic symbols onto the user's view of the environment in real time. Orientation of the user to the environment is handled automatically by a global positioning system (GPS) receiver, a digital compass, and a digital Abney level. Employing a video camera as an imaging device, the user can scan the environment and choose to overlay selected features of any view with cartographic symbols. The video image provides a graphical index to a GIS database of the scene, stored on a portable computer. The user selects geographic objects for augmentation by clicking on them with a mouse or another graphic pick device.

To see how augmented scenes might be implemented and used, we will

- define augmented scenes,

- suggest some application domains,
- outline the steps required to construct and use them,
- determine the effects of positional and directional errors,
- explore the underlying GIS data structure, and
- view the overall construction of a prototype system.

DEFINITION OF AUGMENTED SCENE

An augmented scene is an interactive, symbolized, perspective view of the user's environment seen from his or her current field position that serves as a graphical index to an underlying spatial database. The view may be acquired from captured imagery or simulated with a perspective rendering of the associated 3D surface model. The primary goal of the augmented scene is to provide the user with a direct experience of the map and the environment as a single entity and thereby simplify navigation and map-based queries.

The overlay of acquired imagery on surface models has long been a standard component of computer-assisted map revision systems. Horn (1978) describes a method for registering satellite imagery to surface models for planimetric viewing projections. Drasic and Milgram (1991) discuss techniques for overlaying a user-controlled pointer on captured stereoscopic video imagery to locate objects in the 3D world for robotic navigation within lab settings. This paper extends the use of image acquisition and overlay to perspective views of the general environment.

System overview

The author is currently developing a prototype augmented scene system as illustrated in Figure 1a. Briefly, an augmented scene is created by taking the user's field position (provided by a GPS receiver), horizontal look direction (from a digital compass), and vertical look direction (from a digital Abney level) and combining it with the current focal length and field of view of the imaging device to create a perspective model of the underlying digital elevation model (DEM) registered to the user's actual view. Upon the user's request, the imaging device (in this case a standard video camcorder) captures a single image (frame) of the environment. By default, the computer transmits the frame back to the ocular (or viewfinder) of the imaging device and overlays it with an interactive pointer icon. The user manipulates the pointer in the manner of a mouse or joystick to select a pixel or group of pixels in the image for symbolization (Figure 1b). The computer uses the position of the pointer to look up features in the spatial database located at the world coordinates represented by the selected display coordinates. If poor environmental conditions prevent the acquisition of visual data, the system will use the known positional, directional, and optical parameters to display a perspective map of symbols representing user-selected themes within the current viewing region. A detailed discussion of these procedures

follows in the section "Constructing an Augmented Scene."

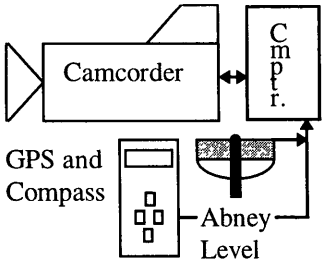


Figure 1a. Augmented scene system components.

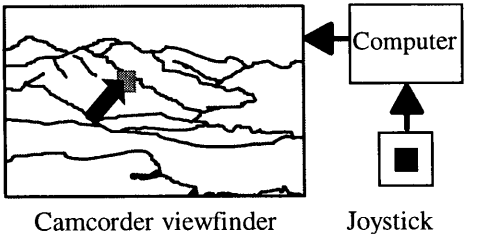


Figure 1b. View through system viewfinder. Joystick moves selection pointer.

APPLICATIONS OF AUGMENTED SCENES

Having developed a definition and structural description of augmented scenes, we can consider some of their potential applications. A few of the most interesting applications include:

Location of utility infrastructure. Utility workers could establish an initial framework for construction or repair of underground conduits by creating an augmented scene from a ground position. The scene would overlay symbolic representations of utility features on an acquired image.

Forest fire management. Teams could query the system for evacuation routes under heavy smoke conditions. Users would see a fully-symbolized perspective view of their environment when smoke conditions preclude the acquisition of useful visual information. They would require the ability to "look behind" objects to navigate beyond their current viewshed.

Recreational uses. Hikers could query the system for the names of features in their environment and their distance to them. Their primary interest will likely be the determination of names and distances to visible features in their environment. They may also ask, "What's behind that hill over there?"

Urban applications. Given a sufficiently rich urban GIS, users could query a building within an image for ownership, structural information, and escape routes. This would require the augmented scene to shift the virtual viewing position anywhere along the user's line of sight.

CONSTRUCTING AN AUGMENTED SCENE

The construction of an augmented scene consists of 1) determining the user's position, horizontal look direction, and vertical look direction; 2) determining lens parameters of the imaging device; 3) building a perspective model of the landscape within the scene; 4) capturing a representative frame from the video camera; 5) registering the captured frame with the perspective model; and 6) transmitting the computer display of the frame (including the mouse pointer and menu structure) to the imaging device ocular.

1) Establishing user position and view angles. The user's position is acquired through a GPS receiver that transmits its data to the system computer. GPS is the most suitable technology for this project because it is highly portable and, under real-time differential operation, accurate to within several meters of true ground position.

Urban environments pose well-known problems for GPS signal reception. The accuracy of a positional calculation depends upon the clear, straight-line reception of timing signals from 4 or more satellites. Buildings can block signals entirely or reflect them, artificially increasing the travel time of the signal to the receiver (Leick 1995). For now, we must allow the manual input of user world coordinates where a GPS solution is not available.

The user's horizontal look direction is determined by a digital magnetic compass which sends its output to the system computer. Like a standard magnetic compass, a digital compass measures the angle to a point relative to the user's magnetic meridian. The magnetic bearing is typically corrected to compensate for magnetic declination. The accuracy of digital compasses is comparable to other magnetic compasses and subject to the same types of error generated by local magnetic disturbances or improper handling.

The vertical look direction can be measured with an Abney level built to provide its data in digital format. At the time of this writing, the author was unable to find a commercially-available, digital Abney level. Since digital theodolites are commonly used to measure both horizontal and vertical angles, it is reasonable to assume that the much simpler Abney level could be produced in a digital format as well. For this project, we will assume that digital input is available but the prototype will require the user to input the vertical angle manually.

2) Determining lens parameters. The prototype system uses a video camcorder as the imaging device. Most standard consumer-grade video camcorders allow the user to change lens focal length and field of view in a single zoom operation. Unfortunately, none but the largest and most expensive video cameras provide digital information about these parameters to an external source. If we

know the number of video frames that occur between a fixed-speed zoom from the shortest to the longest focal length, we can make a very rough estimate of the current focal length as a function of time from one of the two extremes. The project prototype will not attempt to provide this type of zoom capability but will instead limit imaging to the known shortest and longest focal lengths for the project camcorder.

3) Building a perspective model. Once the user's position, horizontal look direction, and vertical look direction are established, we must create a viewing model for the DEM that matches these parameters and incorporates information about the current focal length and field of view of the video camera lens.

Elevations in a DEM are measured with respect to the surface of the ellipsoid specified by a given datum. Since the surface of the ellipsoid is curved, the height of distant objects will appear lower than close objects of the same height, beyond corrections for perspective scale reduction. Equation 1 is a standard correction applied to elevations read from a staff with a surveyor's level in a geodetic survey (Bannister, Raymond, and Baker 1992). It is used here to lower the reported height of a sample in the DEM as a function of its distance from the viewpoint. In Equation 1, z is the original elevation reported in the DEM in meters, d is distance between the viewpoint and the object in kilometers, and $z_{correct}$ is the adjusted elevation for the sample in meters. Additional error in the perceived height of an object due to atmospheric refraction are not taken into account in Equation 1.

$$Z_{correct} = Z - (.078d^2) \quad (1)$$

A perspective projection is created by first transforming 3D world coordinates (elevation values registered to a planar coordinate system such as UTM) to a 3D rectangular eye coordinate system in which the Z axis is collinear with the user's line of sight. Visual perspective is simulated by scaling eye coordinates with respect to their distance from the view point.

3D rendering libraries provide these transformations and allow the user to specify the focal length and field of view of the "camera" representing the user's point of view. By providing the lens parameters of the actual video camera to the rendering functions, the augmented scene can scale the perspective view of the DEM to the video image in the camcorder viewfinder.

4) Capturing a representative frame. For most applications, an augmented scene system needs to capture an individual frame representing the user's viewshed and overlay an interactive mouse pointer and cartographic symbols. The prototype uses a video capture chip on the system computer to read a video sig-

nal from the camcorder and load it into the frame buffer. An augmented scene only requires the capture of individual frames (video stills).

5) Registering the frame with the perspective model. Since the lens parameters of the video camcorder match those of the DEM perspective model, the captured image and the model should be registered if the user's position and view angles are correct. To ensure that overlain symbols will fit the surface of the captured image exactly, the image can be applied as a texture wrap over the DEM perspective model. Rendering libraries use texture wrapping functions to take a 2D image (the captured frame) and drape it over the surface of a 3D object (the perspective model).

6) Transmitting the computer display of the frame to the imaging device ocular. Once the image has been captured into the frame buffer of the system computer, the augmented scene software must transmit the image back to the video camera viewfinder along with its menu structure and the mouse pointer. The prototype accomplishes this by simply redirecting its video monitor output to the camcorder.

VIEWING AND QUERYING AN AUGMENTED SCENE

After the construction of an augmented scene, the user is presented with an image of the landscape which he or she can use as an index to geographic information. By moving a graphics pointer onto a part of the image and clicking, the user can ask questions like, "What am I looking at?" or "How far is that fire tower from here?" To support these queries, the system must 1) allow the user to select an object theme and an operation to act upon it; 2) reverse project the 2D image coordinates of a selected pixel to establish a vector in the world coordinate system parallel to the eye Z axis; 3) allow the user to select one of the set of surfaces intersecting the eye Z vector; 4) symbolize a feature at the selected location if it is within the user's viewshed, otherwise render the view along the vector as seen from the selected surface; and 5) pan and zoom on the current viewshed.

1) Selecting a theme and an operation to act upon it. The interactive operation of an augmented scene begins when the user selects a theme for querying and displaying. Thematic operations are selected by clicking on items in a menu system. A hiker may elect to activate a **Campsite** theme and select the **Display** operator that will symbolize all campsites in the viewshed. If he or she then selects the **Distance** operator, the straight-line distance from the current location to subsequently selected campsites (or the ground distance along a path) will appear in a text window.

2) Reverse projecting the 2D image coordinates. To make this type of selection work, a mouse click in the image must generate the 3D world coordinates associated with the selected pixel. Most 3D graphics rendering libraries provide functions to reverse project image coordinates to world coordinate space. However, a single pixel in a 2D scene maps to a 3D vector passing through the horizontal and vertical coordinates of the pixel and orthogonal to the plane of the viewing screen (Figure 2). This is similar to a view of a tree that we might see out of the window of a house. Branches that appear to cross one another are, of course, at different distances from the window but project, at the area of apparent overlap, to the same region of 2D coordinate pairs on the glass. If we want to identify one of the overlapping branches, we would simply indicate "the closer one" or "the further one."

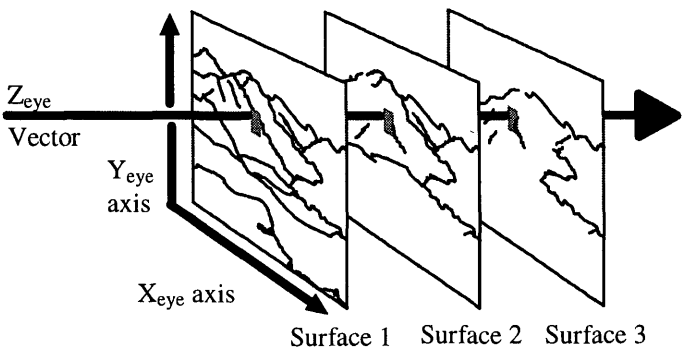


Figure 2. A selected pixel in 2D screen space represents an infinite number of points along the Z vector in eye space. The Z vector may intersect a number of surfaces in the perspective model.

3) Selecting a surface. An augmented scene must provide criteria for selecting which of the infinite points along the eye Z vector is to be referenced by a given pixel. The user may only want to see information on objects that are visible from the current viewing position. Alternatively, it may be important to see information on obscured objects. 3D rendering libraries handle these issues by allowing the user to select one of the set of objects "covered" by a given pixel. The hiker looking for a trail on a hill slope doesn't need (and doesn't care) to see a representation of the hill substructure. Yet the same hiker might ask, "What's on the other side of the hill?" or perhaps more precisely, "What is the view in this direction from the other side of the hill?" We encounter the same types of queries in urban environments. Will the user want to know the name and owner of a building or an elevator shaft inside of it?

The prototype uses the Microsoft DirectX graphics libraries for image modeling and rendering support. DirectX provides a library function that creates

a list of surfaces, sorted by depth from the viewpoint, that project to a selected pixel. By default, the prototype models (and renders, if the system cannot capture a video image) only the surfaces in the user's current viewshed. If the user right-clicks on a pixel, the user's point of view will shift to a location on top of the next surface in the list at the selected pixel coordinates. The viewing parameters that were used to create the initial surface will be used again to render the new viewshed from the updated viewpoint.

4) Symbolizing features of selected themes. Symbols in an augmented scene refers to any graphic marks other than the background video frame or rendered surface. They highlight visible features captured in the current video frame or locate features on a rendered surface (for which no frame exists).

Selected objects are symbolized by opaque, 2D polygons defined as closed lists of vertices and faces. Predefined symbols will typically represent objects in the current theme though users will have the option of changing the default symbol sets. Rendering library functions overlay symbols on the video frame or rendered surface. Whenever a user deletes a symbol, the windowing system redraws any underlying graphics.

5) Panning and zooming on a surface. The user may elect to change the scale of the image (zoom) or move the image viewpoint (pan) to any location. The DEM will be rendered over any portions of the new view not covered by the current video frame.

DETERMINING THE EFFECTS OF POSITIONAL AND DIRECTIONAL ERRORS

Positional accuracy in an augmented scene is most critical for areas in the viewshed nearest the user. If the user wants to see an image of the pipes extending from immediately below his or her feet to the edge of the viewshed, positional error will be most visible in objects projected onto the scene nearest the viewpoint. As objects recede from the viewer, perspective scale reduction will continue to decrease the distances between their visible representation in the captured image and their projected locations in the perspective model.

In contrast to positional error, error in the look direction angles becomes more critical as distance increases away from the user. At a distance of 1 kilometer, a 1° error in either look direction will offset a feature approximately 17.5 meters in 3D world space from its true position. At a distance of 10 kilometers, the error will be approximately 175 meters.

DESIGN OF THE DATA STRUCTURE

The data for an augmented scene will consist of a DEM and a set of the-

matic coverages. To calculate the amount of data that will be required, we need to establish a sampling density and a maximal viewshed over which the system must operate.

Using Equation 1, we find that an object rising 500 meters above a viewpoint has a corrected elevation of 0 meters at a distance of approximately 80 km from the viewpoint. Does this mean that we would require DEM and thematic data over a radius of 80 km away from this point? Most users will probably show little interest in features that are close to disappearing below their horizon, yet some (like a driver on a highway) may want to know the name of the mountain range that just appeared above it. Assuming a 30 meter grid sampling interval, we would need approximately 22,000,000 DEM grid cells alone to cover a circle of this radius. A 15 meter interval will require 4 times this amount. Visible objects rising higher than 500 meters above the viewpoint will extend the viewshed boundary and the required DEM coverage even further. Although data sets of this size can easily be stored on current CD-ROM media, constructing a perspective model for a large viewshed may require prohibitive amounts of computing time.

To solve this problem, an augmented scene must either limit its working range or build perspective models by resampling data at increasing intervals with distance from the viewpoint. In a perspective view, the ground area covered by a unit area of the viewing screen increases with the distance of the ground from the screen. Pixels representing areas at large distances from the screen can cover many cells in the database. Therefore, information about distant features can be resampled at a greater interval than the default sampling interval of the DEM. Resampling can be handled as a continuous function of distance from the viewpoint or by establishing a single threshold distance between two sampling interval regions.

Most rendering libraries use the concept of clipping planes (parallel to the viewing screen) to identify visible regions within a viewing pyramid extending from the user's eye. Only the region in the viewing pyramid between the front clipping plane (near the viewer's eye) and the back clipping plane (far from the viewer's eye) is rendered. Instead of using the back plane for clipping, the system uses its location to separate regions of higher and lower resolution sampling. The default location is based on the user's preference with regard to system processing speed and memory capacity. The user can move the default location of the back clipping plane to handle specific imaging requirements.

BUILDING THE PROTOTYPE SYSTEM

The prototype for this project uses an IBM ThinkPad 755CD as the system computer. The 755CD uses an on-board video capture chip to receive input from a Sony CCD-TRV70 camcorder. The camcorder also receives video output from

the 755CD to see the captured image superimposed with menu and pointer data. A Silva GPS Compass provides both positional and horizontal look direction data. Data from the GPS Compass is transmitted to the 755CD over through a standard serial communication port. The system software is composed of a main program, Urhere, written by the author in C++ for the Microsoft Windows 95 operating system.

CONCLUSION

The intent of this paper has been to define augmented scenes, suggest some applications for them, and to propose a model for their construction and use. The most important topics for future research includes:

- reducing positional and directional error;
- reducing the size of data sets;
- incorporating non-perspective (plan) views;
- determining a suitable data distribution format; and
- refining the user interface through subject testing.

It is expected that technological developments in portable computing and position finding devices will greatly increase the feasibility of augmented scenes in the near future.

REFERENCES

Bannister, A., S. Raymond, and R. Baker (1992). *Surveying, 6th ed.*. Longman Scientific and Technical, Essex, England, pp. 76-79.

Drasic, D. and P. Milgram (1991). Positioning accuracy of virtual stereographic pointer in a real stereoscopic video world. In: *SPIE Stereoscopic Displays and Applications II*. Vol. 1457, pp. 302-312.

Horn, B. and B. Bachman (1978). Using synthetic images to register real images with surface models. In: *Communications of the ACM*. Vol. 21 No. 11, pp. 915-924.

Leick A. (1995). *GPS Satellite Surveying, 2nd ed.* John Wiley & Sons, New York, pp. 311-313.

IMPROVING MOVING MAPS: A SYSTEM FOR FEATURE SELECTION BASED ON A NEW COGNITIVE MODEL

Paul Van Zuyle
Dept. of Geography
University of California
Santa Barbara, CA 93106
vanzuyle@ncgia.ucsb.edu

ABSTRACT

This paper describes a system for selecting objects for display on moving maps from a cartographic database. A cognitive model similar to spatial interaction theory is used to predict the most salient objects for display. Animated maps have been created using concepts described here.

INTRODUCTION

Electronic charting systems offer virtually unlimited opportunity for presenting spatial information to navigators. At the same time, much of their potential is often lost because information inappropriate to the task at hand clutters the screen, or important data is omitted. While there are many dimensions to this problem, the one addressed here is this: given a database of cartographic objects, which ones should be displayed to best serve the navigator?

Cartographic generalization in GIS can be considered in three distinct phases; object generalization, model generalization, and cartographic generalization (McMaster & Shea, 1992). The system described here falls into the domain of model generalization, which Weibel (1995) identifies as the step most amenable to formal modeling. That is to say it makes no attempt to determine the appropriate data model, or to determine the appearance of individual objects on the output device. Instead, this system is designed to facilitate data reduction and control how many and what type of objects are chosen for display.

Cognitive properties of moving map displays have received some attention from the aviation research community (e.g. Aretz, 1991). That attention has focused primarily, however, on map orientation rather than content.

Cartographers, on the other hand, have spent considerable effort on automating the procedures of generalization to control the appearance and content of maps. But this effort has not yet come close to reproducing the results of human cartographers, and some of it may be misguided in its attempt to reproduce traditional cartographic products (Goodchild, 1988).

In contrast, the maps displayed by the system described here change continuously. They are updated based on a number of parameters such as velocity, scale, and other inputs which will be described below. This reflects a philosophy that the information presented should be useful in a specific situation, rather than be a comprehensive reference.

The model described here is being implemented so that users can adjust parameters and see results quickly. That should facilitate the design of experiments which will attempt to test the validity of the cognitive model.

METHOD

An examination of a series of aeronautical charts at different scales shows a variety of differences. Scale, symbology, level of detail and density of selected objects all vary. In addition, there is a systematic change in the emphasis placed on different types of objects with a change in scale. This makes sense, since pilots are engaged in different kinds of tasks when they consult different charts. Small-scale flight planning charts and those used at high altitude display mostly airport and airway information. Large-scale charts used for instrument approaches, however, have more information about objects that are immediate hazards, such as mountains and high terrain, but less overall density of detail.

The approach taken here attempts to predict the best objects to display by considering three things--the class (or type) of each object in the database, the relative prominence of each object within its class, and an evaluation distance.

In order to illustrate the concept, data from the Digital Chart of the World (DCW) was chosen to portray a simulated aeronautical moving map display. In one sense this is an ideal source, since the DCW was digitized from the Operational Navigation Charts (ONC) produced by the Defense Mapping Agency (DMA). The DCW, however, lacks important detail because it was originally generalized at a single scale (1:1,000,000).

Each object in a test database of airports and mountains has been assigned an initial importance value and a distance decay coefficient. They each present an opportunity for spatial interaction (positively and negatively, respectively!). In the case of airports, length of the longest runway was used for initial importance, while peak elevation was used for mountains. By chance, these numbers are roughly equivalent across the two classes. Because mountains are likely to be more important to pilots than airports at short distances, and less important when far away, they are assigned a steeper distance decay coefficient

than airports. The following equation, a standard formula for halving distance, was used:

$$I = Z2^{-R/B}$$

where:
I = importance, used to determine which objects will be selected for display.
Z = initial importance
R = evaluation distance
B = distance decay rate

This formula is used to generate a score for each object in the database. Then the top n objects are displayed, where n is a number preselected as a control for map clutter. When R is set to a large value, airports tend to be selected in preference to mountains, since distance decay is set lower for airports, and the resulting scores are higher.

The result is that the map display tends to show a preponderance of airports or mountain peaks, depending on the value of R chosen. While in this equation R is independent of scale, a logical implementation would make R a function of scale.

These rules have been encoded in ArcView 3 so that different values of each of the parameters (along with scale) can be modified by the user.

RESULTS

A set of animated maps has been prepared that demonstrates the effect of changing R, along with differences in n and scale. They can be viewed at http://www.ncgia.ucsb.edu/~vanzuyle. One characteristic that is noticeable in the animations is that some of the objects tend to appear and disappear multiple times with a continuous change of scale. This is a consequence of the scoring system using multiple parameters.

Another apparent difficulty is label placement. A better label placement algorithm is required if this is to be a practical implementation. A system of label placement based on importance level has been demonstrated by Arikawa (Arikawa & Kambayashi, 1991).

The next stage in this research is to experiment with potential users, such as pilots. The aim is to validate the cognitive model and see which of these variables produces significant differences in navigational performance. Eventually, an automated moving map system may be developed that selects the optimal value of R and n, as well the scale, for a given navigational task.

REFERENCES

Aretz, A. 1991. The design of electronic map displays. Human Factors. 33(1):85-101.

Arikawa, M., Kambayashi, Y. 1991. Dynamic name placement functions for interactive map systems. Australian Computer Journal. 23(4):133-147

Goodchild, M. 1988. Stepping over the line: Technological constraints and the new cartography. American Cartographer 15(3):311-319.

Lohrenz, M.C., Trenchard, M.E., Van Zuyle, P., Perniciaro, R., Brown, C., Gendron, M., Myrick, S. 1996. TAMMAC Digital Map Requirements Study in Support of Advanced Cockpit Moving-Map Displays. NRL/FR/7441-96-9652.

McMaster, R.B., Shea, K.S. 1992. Generalization in digital cartography. Washington, D.C.:Association of American geographers.

Weibel, R. 1995. Three essential building blocks for automated generalization. *In* GIS and Generalization. ed. Muller, J.C., Lagrange, J.P., Weibel, R. London: Taylor and Francis.

USING SPACE/TIME TRANSFORMATIONS TO MAP URBANIZATION IN BALTIMORE/WASHINGTON

Lee De Cola, U.S. Geological Survey*
521 National Center, Reston VA 20192, ldecola@usgs.gov

ABSTRACT

During the past year researchers at the U.S. Geological Survey have been using historical maps and digital data for a 168-km × 220-km area of the Baltimore-/Washington region to produce a dynamic database that shows growth of the transportation system and built-up area for 270-meter grid cells for several years between 1792 and 1992. This paper presents results from the development of a *Mathematica* package that spatially generalizes and temporally interpolates these data to produce a smoothly varying urban intensity surface that shows important features of the 200-year urban process. The boxcount fractal dimension of a power-2 grid pyramid was used to determine the most appropriate level of spatial generalization. Temporal interpolation was then used to predict urban intensity for 4320-m cells for 10-year periods from 1800 to 1990. These estimations were spatially interpolated to produce a 1080-m grid field that is animated as a surface and as an isopleth (contour) map (see USGS 1997 for the Internet address of the animation). This technique can be used to experiment with future growth scenarios for the region, to map other kinds of land cover change, and even to visualize quite different spatial processes, such as habitat fragmentation due to climate change.

In 1994 a team of U.S. Geological Survey (USGS) and academic researchers produced an animation of the growth of the San Francisco/Sacramento region using a temporal database extracted from historical maps, USGS topographic maps, digital data, and Landsat imagery (Gaydos and Acevedo 1995). Publicly televised videotapes of this work received sufficient attention to support a larger team that had planned to work on the development the Boston/Washington megalopolis (Gottmann 1990) (The

*Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

current research involves staff from USGS, National Air and Space Administration, the Smithsonian Institution, and University of Maryland Baltimore County.) Resource and time constraints, however, limited efforts to the southern part of the region shown in figure 1 (Crawford-Tilley et al 1996, Clark et al. 1996). The animation of urbanization in this region is based on a 512^2-cell grid data structure that represents whether or not a given 270-meter cell is built-up in each of 8 base years (figure 2). This raster was interpolated for intervening years, but still represents a binary condition for each of the grid cells. Throughout this work there was interest in how we might analyze the intensity of development, perhaps by sacrificing spatial resolution for temporal and measurement resolution (table 1). Because the urban phenomenon (cartographic feature) is self-organized, complex, and probably also critical (Bak 1996), it is reasonable to suppose that scaling properties would assist in this transformation (Quattrochi and Goodchild 1997).

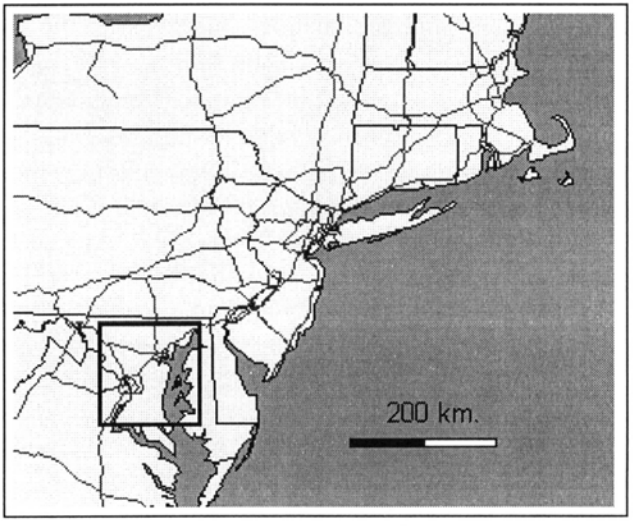


Figure 1. The study area in the Boston-Washington megalopolis.

Table 1. Dimensions of the data

DIMENSION		EXTENT	RESOLUTION *l*	
			DATA	ANIMATION
SPACE *x*	Balt/Wash	138 km	270-meter grid	1080-meter
TIME *t*	1772-1992	200 years	~25 years	10 years
FEATURE *f*	Land cover	Built-up	[0, 1] binary	[0, 256]

Consider therefore a location in space x at a given time t and spatial resolution level l for which a measurement f is made; call this measurement $f_l(x_t)$. For example, in the present case we are interested in whether or not a given grid cell of a certain size is built-up (covered by buildings, has a dense road network, etc.). In this simplest case we have a binary function $f_l(x_t)$ = {1 if x_t is built-up, 0 otherwise}. Assume at the finest scale level $l = 0$ that this measurement is reliable—but what can be said of the phenomenon at other spatial scales? Table 2 shows how a 10-level power-2 image pyramid can be built upon the 0-level data in the present case. One (not necessarily obvious) way to examine data at coarser scales is simply $fbox_{l+1}(x_t)$ = {0 if all $f_l(x_t) = 0$, 1 otherwise}, i.e. the value of a higher-level $l + 1$ cell will be "on" if any of the lower-level l cells is on. This is called a box-covering algorithm because a high-level box is needed to cover 1 or more lower-level boxes (De Cola 1997). Consider the 0-level image of figure 2, which contains 41,183 built-up cells, as reported in the last row of table 3, which presents the box counts for each level and each of the raw data years The table shows at the next highest level $l = 1$ that 14,892 cells are necessary to cover these cells. This number is 45% larger than the (41,183 / 4 =) 10,296 level-1 cells that would be necessary if all the level-0 cells were spatially compact. The excess number is due to spatial complexity of the urban phenomenon, which has fractal dimension $D < 2$, where $D = 2$ would be the dimension of say a perfect disk (for a comprehensive discussion of the fractal nature of cities see Batty 1995).

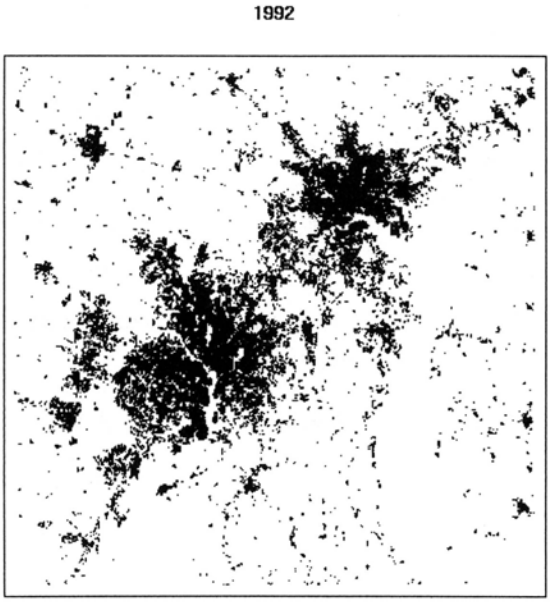


Figure 2. Level 0 grid for 1992.

Table 2. Characteristics of the image pyramid

LEVEL	CELLS PER ROW	CELL SIZE (meters)	MAXIMUM VALUE	EXAMPLE
9	1	138,240	262,144	Size of the study area
8	2	69,120	65,536	
7	4	34,560	16,384	
6	8	17,280	4,096	
5	16	8,640	1,024	
4	32	4,320	256	Interpolation level
3	64	2,160	64	
2	128	1,080	16	Animation
1	256	540	4	
0	512	270	1	BaltWash Pixel

The 0-level row of the table 3 illustrates that for at least 200 years there has been some urbanization in the region (A fit of a linear model to the 0-level data yields $\ln[f_0(x_t)] = -40 + 0.026\ t$ which predicts a y-intercept at about the year 1575). The table cells that are shaded represent completely covered pyramid levels, showing how in later years the windows rapidly become saturated. This happens at $l = 8$ in 1792 and by level 6 in 1972 and later. One way to avoid this saturation is to expand the extent of the study area, and this indeed is underway. But another problem with this analysis is that traditional maps (1772-1850) produced to widely varying cartographic styles, are being analyzed along with carefully standardized USGS maps (1900-1953) and satellite imagery (1972-1992). Nevertheless—and this is another advantage of multiscale analysis—at coarser scales the difference among these disparate data sources diminishes.

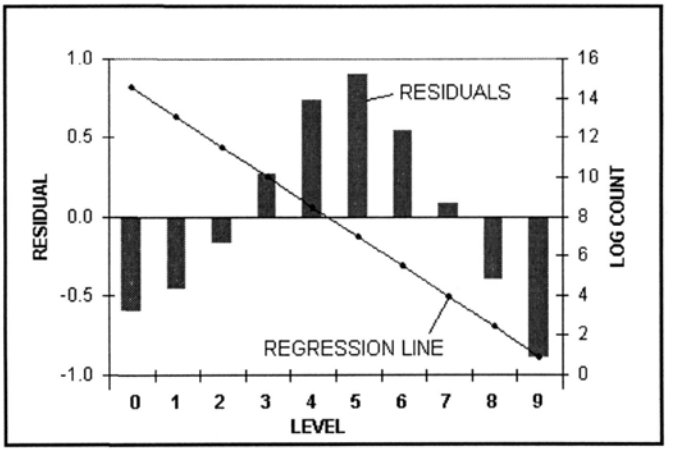


Figure 3. Fractal dimension estimation 1953.

Table 3. Box Counts for each year and level

LEVEL	1792	1850	1900	1925	1953	1972	1982	1992
9	1	1	1	1	1	1	1	1
8	4	4	4	4	4	4	4	4
7	15	16	16	16	16	16	16	16
6	31	42	62	63	63	64	64	64
5	40	75	190	218	230	242	243	245
4	52	99	360	457	588	750	770	784
3	59	126	539	763	1216	1894	1985	2076
2	83	197	909	1402	2560	4741	5118	5448
1	142	412	1790	2897	5956	12296	13564	14892
0	286	1069	4431	7089	15463	33092	36742	41183

The box counts in table 3 can be used to compute the fractal dimension of the built-up area for each year. For example, figure 3 shows the regression line estimating $\log_2[f(x_{1953})] = 0.89 - 1.51\ l$ for 1953, which yields a fractal dimension of $D_{1953} = 1.51$ and an $R^2 = .99$ (Falconer 1990). The box counts for each level and each year are used to compute the 8 values of D_t, the fractal dimensions for each of the data years, shown in figure 4. There is a continuing debate in urban studies about how regions develop. One school argues that so-called "primate" metropolitan regions continue to grow from a point to a centralized but spreading metropolitan pole. But another school envisions a dispersed metropolis that may eventually completely disperse, returning to a collection of isolated points (Alonso 1980, De Cola 1985). Figure 3 certainly shows the early stages of this process; we can only speculate about whether D_t will eventually decline, although its rate of increase seems to be leveling off. This scenario suggests the possibility of future dispersion in which the urban complex not only breaks up into dispersed centers but even perhaps returns to the low-dimension post-industrial "village" system similar to that of the 18th century.

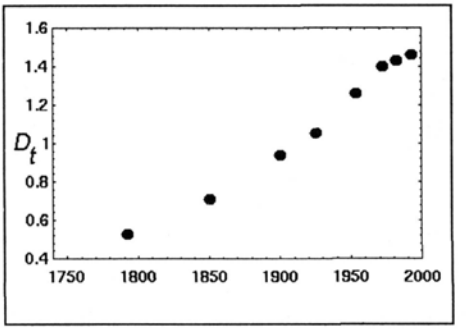


Figure 4. Boxcount fractal dimensions 1792-1992.

Each of the fractal dimensions D_t for the data years is a linear estimate of the behavior of the box counts over the scale levels. Yet the fit is not perfect, as figure 3 shows for 1953; there is a similar pattern of parabolic residuals among all the years. In general the middle scale levels $l = 4$ and 5 have higher residuals, suggesting that at about the 6-km scale the urban area has its most compact representation. But the box count aggregation algorithm, which yields 0/1 values, cannot be used to generalize the data. Another way to aggregate grid data is to sum lower-level values using $fsum_{l+1}(x_t) = \Sigma f_l(x_t)$ where the aggregation is over subwindows of 4 cells each. The algorithm *fsum* is like a mean filter that aggregates subregions into a higher-level region whose value is the average of lower-level elements. The generalized animation is therefore based on the level-4 generalization, which gives for each of $32^2 = 1024$ cells of size 4320-m an 8-bit dynamic range of [0, 256] (see table 2). Figure 5 shows what happens to the 1992 data for 5 successive levels of aggregation. The lower-level images allow us to focus on the individual features of the region, while the higher-level images highlight the unified nature of the BaltWash metropolis.

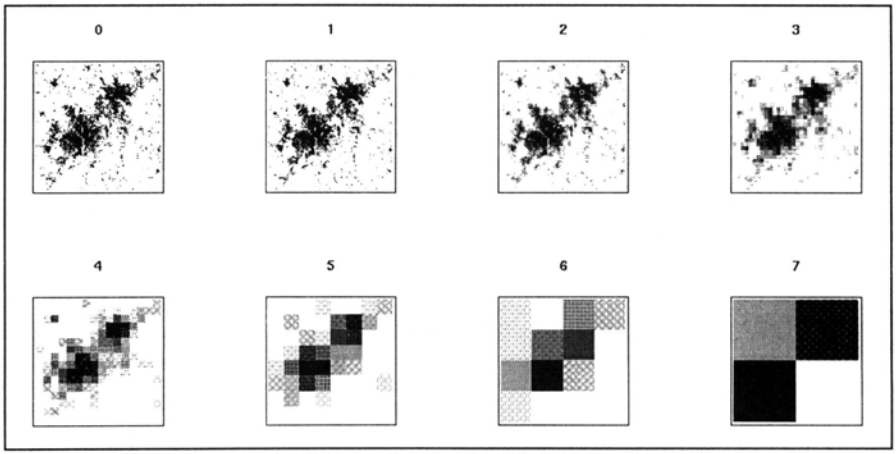


Figure 5. Sum pyramid for 1992.

Let $l = 4$ and consider the central-cell x = (col, row) = (16, 16) for each of the $t = 1,\ldots,8$ data years. The values of $fsum_4(x_t)$ for this cell are shown in figure 6 and (as did D_t in figure 4) these points suggest a logistic curve, which can be estimated with an interpolation (prediction) function $fsum^p$ that predicts *fsum* for any year and not just the 8 data years. Figure 6 shows $\{fsum^p{}_4(x_t)$: $x = (16,16)$, $t \in [1750 \text{ to } 2000]\}$. When this function is used at level-4 we only get 32^2 predictions. This is how we obtain a gain in feature resolution (from [0,1] data to [0, 256] values), and a gain in temporal resolution (from 8 irregularly spaced

measurements to 20 decadal interpolations), by sacrificing a loss in spatial resolution (from 270-m to 4321-m cells).

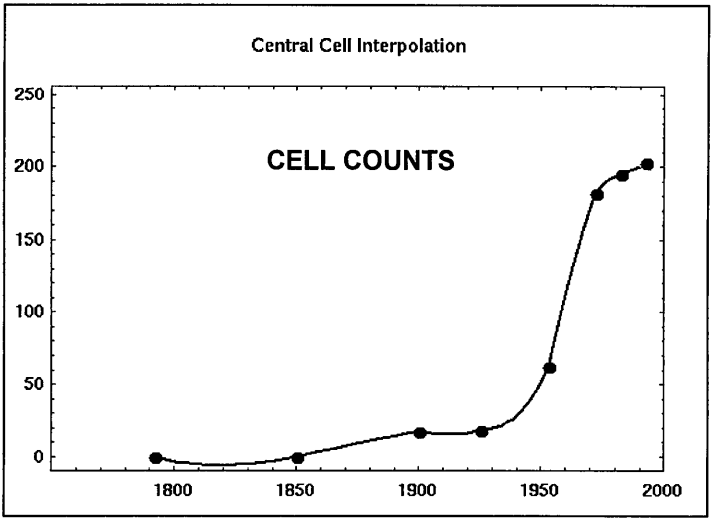


Figure 6. Actual and interpolated values for cell (16,16).

The unique temporal interpolation functions for each of the (32^2 =) 1024 level-4 cells can be arrayed into a *Mathematica* table that provides a grid of predictions for any year in the study period. A sample for 1990 is shown in figure 7, taken from the animation (USGS 1997). The data have been spatially linearly interpolated to level 1 (540 meters) to provide a smooth surface for visualization (for a alternative approaches to the interpolation problem see Tobler 1979 and Bracken and Martin 1989). The image, which is one frame of a 20-period animation, illustrates the polycentric nature of the Baltimore/Washington urban process. The animation shows reveals a self-organizing system that has been growing along the Northeast U.S. transportation corridor. During the past 200 years urban leadership has shifted between the two centers at least three times, and since World War II there has arisen a polycentric post-industrial system whose fractal dimension has been growing logistically and may be leveling off.

Another way to visualize the growth process is isopleths or contours, which emphasize the geographic location of urbanization. figure 8 shows not only the 2 urban centers in 1992, but such other features as the edge cities of Frederick, Annapolis, and La Plata, MD as well as Potomac Mills, VA. The picture also highlights the linear nature of the whole system, oriented along Interstate 95, which continues from Boston to Miami.

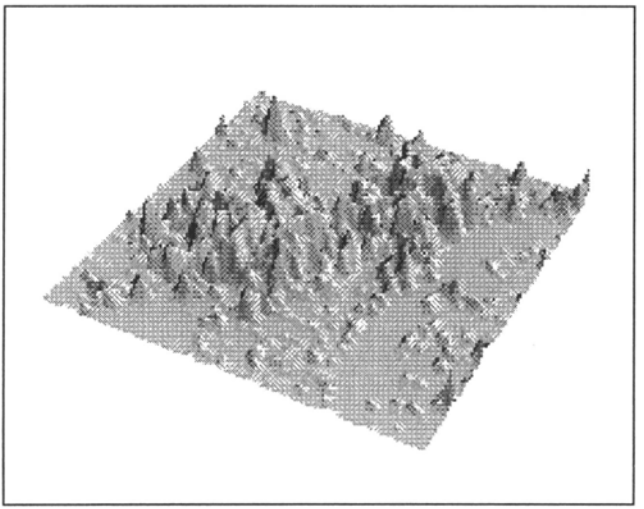


Figure 7. 3-d plot of estimated built-up areas for 1990.

Naturally we are interested in the future of the region, and the analysis suggests approaches. (A logistic curve fitted to the 0-level data in table 2 yields $f_0(x_t) = 55800\ [1 + \mathrm{Exp}(2.09 - 0.0469(t - 1923))]^{-1}$, which has a maximum growth rate of 2.1% in 1923 (Haggett, Cliff and Frey 1977:238)). This expression has an asymptotic value of 55,800 pixels, which is only about 20% of the window at level-0.

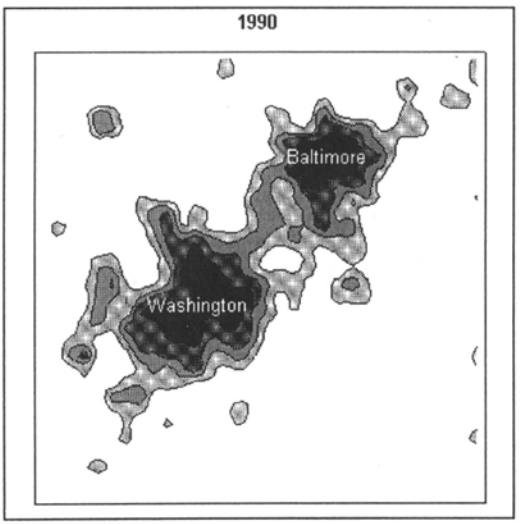


Figure 8. Contour plot of estimated built-up areas 1990.

The analysis of the last 3 data years (1972, 1982, 1992) was based on Landsat imagery, and the growth both of the fractal dimension D_t (figure 4) as well as of one of the generalized cells $fsum_4(x_t)$ (figure 6) show a linear growth trend. The growth rate for 1972-1992 is mapped in figure 9; darker shades show faster growth—up to 2% per year. Recent metropolitan development displays the doughnut patterns typical of U.S. cities (Whyte 1968). The Baltimore growth ring is broken by Patapsco Bay and the Washington ring by a Potomac River "greenbelt" that would clearly be the fastest growing edge city were the river bridged from Sugarland Run VA to Seneca Creek MD. It is interesting how strongly topography still influences the development of this region.

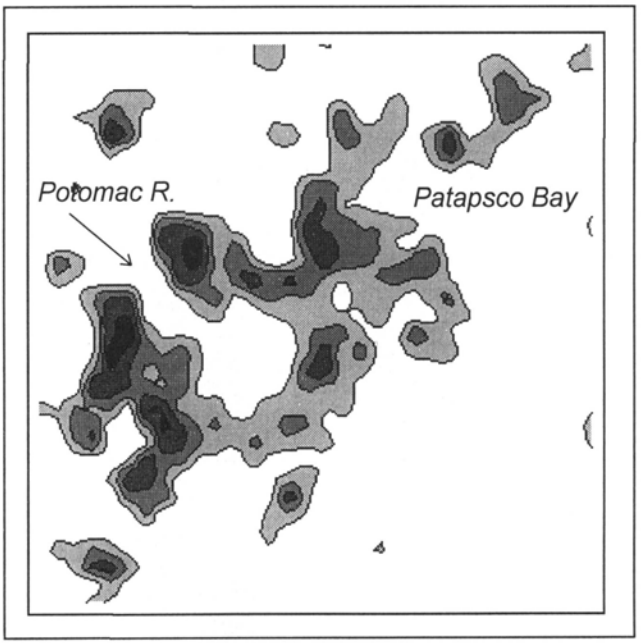


Figure 9. Contour plot of growth rates, 1972-1992.

The research presented here is part of a 118-year history of the use of USGS core skills in the physical, and—more recently—human and biological sciences to understand human-induced land transformations. These efforts exhibit not only institutional expertise but also rich historical databases that can be used to understand spatial processes, to forecast change, and help to shape future policy. The dimensions highlighted in table 1 suggest new directions for this research. First, the analysis can profit from a broader spatial view, expanding to Megalopolitan and even world urbanization. Second temporal extrapolation and deeper "data mining" will help planners envision the future of the region—as well as its distant past. Third, more features (shoreline, land cover, climate) need to be studied and animated. A central theoretical and policy

problem highlighted by this work therefore is the development of rigorous, informative, and visually effective transformations of data along and among spatial, temporal, and phenomenological scales.

REFERENCES

Acevedo, William, Timothy Foresman, and Janis Buchanan 1996 Origins and philosophy of building a temporal database to examine human transformation processes, ASPRS/ACSM Technical Papers I:148-161.

Alonso, William 1980 Five bell shapes in development, *Papers and proceedings of the Regional Science Association*, 45:5-16.

Bak, Per 1996 *How nature works*, NY: Cambridge University Press.

Batty, Michael 1994 *Fractal cities*, NY: Wiley.

Bracken, I and D. Martin 1989 The generation of spatial population distributions from census centroid data, *Environment and Planning A*, 21:537-543.

Clark, Susan C., John Starr, William Acevedo, and Carol Solomon 1996 Development of the temporal transportation database for the analysis of urban development in the Baltimore-Washington region, ASPRS/ACSM Technical Papers, 3:77-88.

Crawford-Tilley, Janet S., William Acevedo, Timothy Foresman, and Walter Prince 1996 Developing a temporal database of urban development for the Baltimore/Washington region, ASPRS/ACSM Technical Papers, 3:101-110.

De Cola, Lee 1985 Lognormal estimates of macro-regional city size distributions, 1950-1970 *Environment and Planning A* 17:1637-1652.

De Cola, Lee 1997 Multiresolution covariation among Landsat and AVHRR vegetation indices, in Quattrochi and Goodchild 1997.

Falconer, K.J. 1990 *Fractal geometry: mathematical foundations and applications*, New York: Wiley.

Gaydos, Leonard J and William Acevedo 1995 Using animated cartography to illustrate global change, International Cartographic Association, Barcelona.

Gottmann, Jean 1990 *Since Megalopolis: the urban writings of Jean Gottmann* Baltimore: Johns Hopkins University Press.

Haggett, Peter, Andrew Cliff and Allan Frey 1977 *Locational analysis in human geography* London: Edward Arnold.

Quattrochi, Dale and Michael Goodchild 1997 *Scale in remote sensing and GIS* Boca Raton FL: CRC Press.

USGS1997 The animation is available on the WorldWideWeb at: http://geog.gmu.edu/gess/classes/geog590/gis_internet/ldecola/baltwash/

Tobler, Waldo R. 1979 Smooth pyncnophylactic interpolation for geographical regions *Journal of the American Statistical Association* 74(367):519-530.

Whyte, William H. 1968 *The last landscape*, Ch. 8, NY: Doubleday.

MODELING URBAN DYNAMICS WITH ARTIFICIAL NEURAL NETWORKS AND GIS

Chris Weisner
RAM Mobile Data USA
cwiesner@smtplink.ram.com

David J. Cowen
Department of Geography & Liberal Arts Computing Lab
University of South Carolina, USA
cowend@sc.edu

ABSTRACT

The modeling of dynamic urban systems has been of interest to spatial analysts for the better part of the past four decades, and the development of geographic information systems (GIS) has sparked continued interest in spatial process modeling. Recent research in a number of far reaching disciplines has shown artificial neural networks (ANN) to be powerful tools for modeling many dynamic systems (Vemuri and Rogers, 1994). This research investigated the possibilities for ANN's as spatial analytic tools. To this end, an artificial neural network was linked with a GIS for the purpose of modeling urban growth in sub-regions of a metropolitan area. The validity of the ANN model was tested against a linear regression model. The results of this research support the hypothesis that ANN are in fact useful spatial analytic tools and can be used to accurately model dynamic urban systems.

INTRODUCTION

Modeling and prediction of urban growth have been of interest to researchers for the better part of the past four decades (Chapin and Weiss, 1968, Batty and Longley, 1994). Much of the rationale behind this research was to determine the cause and effect of the urban form on transportation patterns and to use this knowledge for the planning of future transportation networks. Researchers were interested in the potential for computer models to enable the testing of changes in policy and urban resources on transportation networks, and thus the models proposed were deductive in nature.

The model put forth in this research was an inductive approach to the urban modeling problem, and incorporated an artificial neural network (ANN) in conjunction with a geographic information system (GIS) to model a spatio-temporal database of single family residential building permits. The model was based on the assumption that the time of occurrence and magnitude of urban growth in a sub-region of a metropolitan area is a function of the development already occurring in the sub-region and within it's neighboring areas.

METHODOLOGY

The spatial data structure created for this research was an arbitrarily defined tessellation of 2.6 square mile regular hexagons covering the two county Columbia SC study area. The benefits of using a regular hexagon tessellation was that neighborhood relations, shape, size and orientation are held constant throughout the surface. Building permits are indicators of the morphology of the urban landscape Halls, Cowen and Jensen (1994). This study used the single family residential housing units subset of a building permits database for an eleven year period. For this study, the training set included the building permit data from the years 1981 through 1989. The test set contained the data from 1990 and 1991. For this study a hexagon had to have had at least 10 permits issued at least one of the years during the period. This ensured that there was enough training set data for the neural network to find a pattern of development (fig.1).

Based on previous research it was determined that artificial neural networks model the time series of nonlinear dynamic systems by mapping the state of the system at time t, x(t), to some future state, $x(t + \Delta t)$. Chakraborty et al (1992) demonstrated improved results by incorporating the time series of comparable objects (cities) as inputs, and this approach was adapted to this study by including the states of the "neighborhood" (the six surrounding hexagons) with the state of each hexagon as inputs to the ANN model (fig 2). The spatial relationships between the hexagon and its neighbors are built into the structure of the ANN through the arrangement of the input nodes and the weight connections between the input layer and the hidden layer . This arrangement is held constant throughout the study area. This has the effect of defining a regular semi-lattice organization over the entire surface (fig. 3). In terms of the specifics of the dynamic urban system, this is the urban organization argued for by Alexander (1965) in his two part essay "A City is Not a Tree". Few models have adopted this structure, opting instead for a simpler hierarchical tree-like structure.

Since the spatial relationships between neighboring hexagons are hard coded into the ANN structure, one set of network weights for the entire area would not necessarily provide the best model. In fact, the relationship between each hexagon and it's neighbors changes with respect to the central business district (CBD) throughout the study area (Fig. 4). Once the permits were partitioned in space and

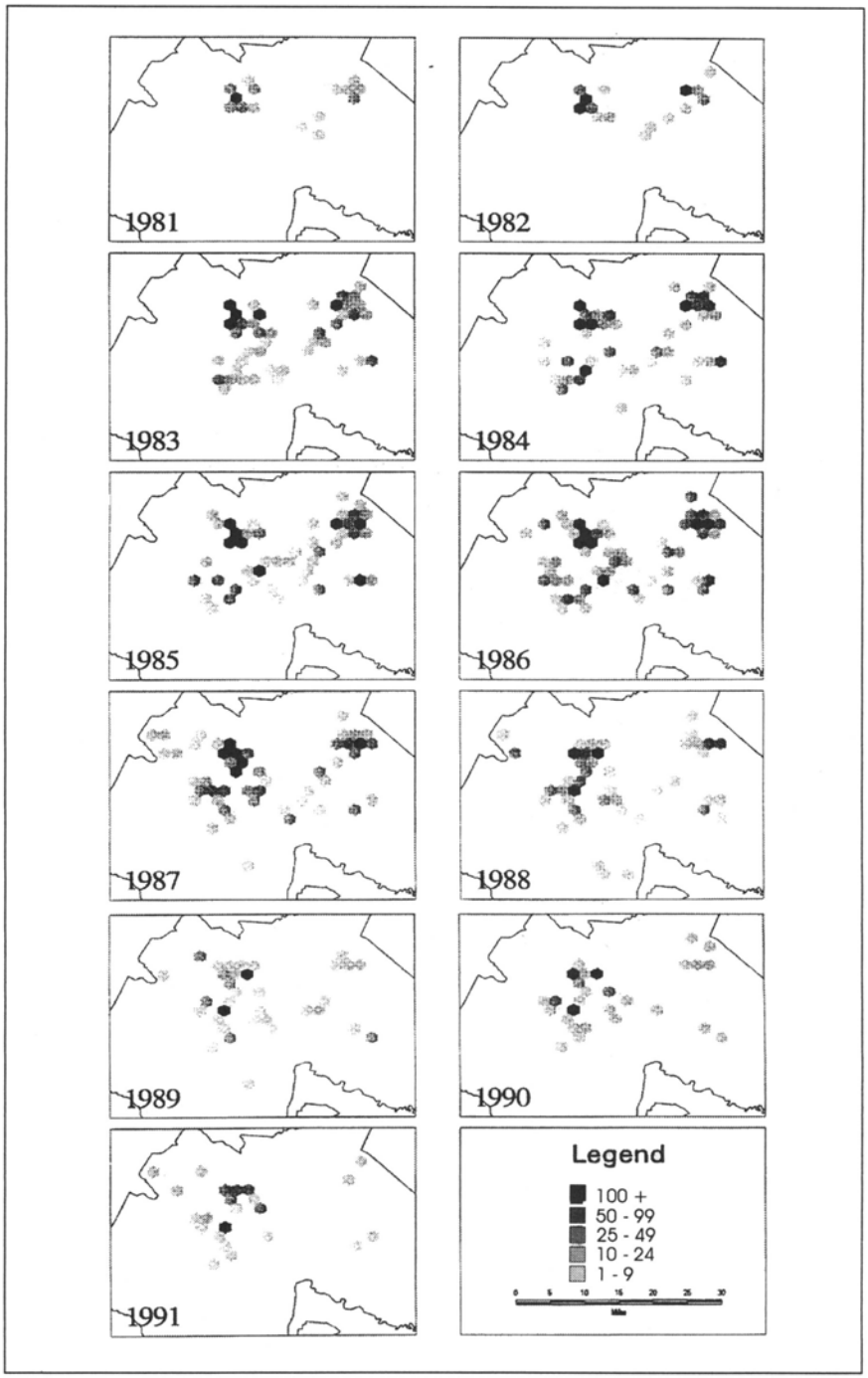


Figure 1 Distribution of building permits in hexagonal data structure.

time, their theoretical time series would start with a period of no growth corresponding to the period when the area was in non-urban land use, a short period of active growth as the urban fringe passes through the area and a final period of no development occurring when the available space in the area has become saturated with development.

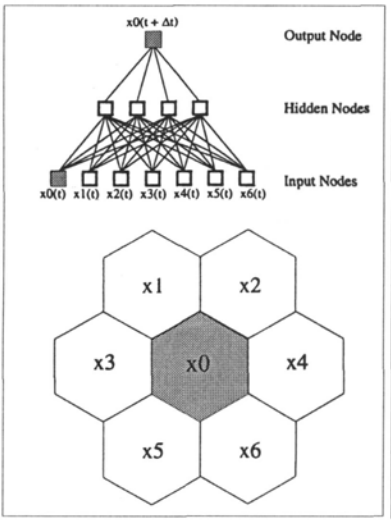


Fig. 2 Hexagonal ANN

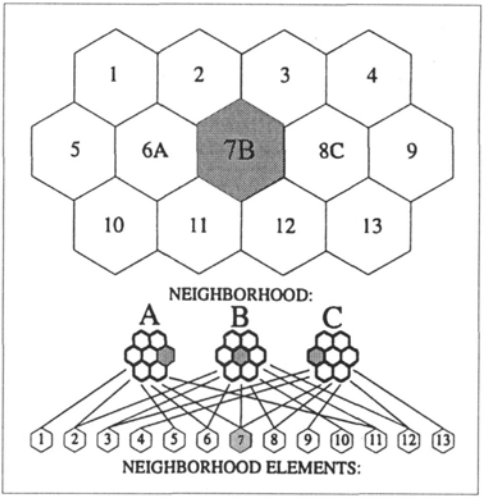


Figure 3 ANN Neighborhood

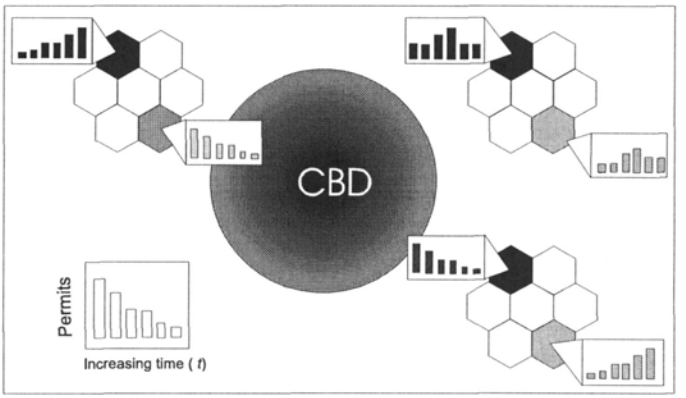


Fig. 4 Hypothetical time series from CBD

The last issue to be resolved was the development of the specifics of the artificial neural network model(s). This process involved the selection of an appropriate number of hidden nodes, the scaling of the data from real world values to ANN values, and the training threshold of the sum of the square error to be used

in training the ANN. A number of researchers have indicated the difficulty of determining appropriate network architecture's (i.e. the number of hidden nodes) for modeling a wide range of data sets (Heermann and Khazenie, 1992; ; Fletcher and Goss, 1993; Lodewyck and Deng, 1993). Lowe and Webb (1991) have suggested that the number of hidden nodes represent a Euclidean dimension into which the dimension of the *attractor* of the system (which may be of fractal dimension) is embedded. For this research, initial feedback indicated that between four and two hidden nodes were adequate to produce acceptable results. Of the ninety-four ANN models chosen, forty-eight (48) utilized four hidden nodes, twenty-five (25) utilized three hidden nodes and twenty-one (21) utilized two hidden nodes. The scaling used in this study incorporated the following rules:

1. The scaling values were between 0.2 and 0.8.
2. The minimum and maximum values for each training set were determined from the center hexagon in the seven hexagon neighborhood. Values in the surrounding six hexagons which were less than the *minimum* were assigned the minimum value and values which were greater than the *maximum* were assigned the maximum value.
3. In many instances the range of activity was still quite large with many small values and a few instances of large values. In these cases, experimentation indicated that taking the log of the data values produced desirable results.

Thus, for each model that was developed, two approaches were used - ordinary linear and log-linear scaling. Twenty-four ANN models were developed for each of the ninety-four hexagons in the study area. The twenty-four models correspond to variations in the number of hidden nodes (2,3,4), the scaling method used (linear, log-linear), and the learning criteria used to end the training phase of the ANN model development. The log-linear scaling method was used by sixty (60) of the ninety-four ANN models and the straight linear scaling was used by thirty-four (34). A final consideration in specifying the model was the learning threshold for the sum of the squared error that must be reached before the training of the network weights ends. It is generally agreed that small sum-of-the-square errors attained during the training phase result in networks which have "learned" the idiosyncrasies of the training data, and may result in poor generalization to the test data set and other data the network has not previously "seen". To adjust for this, this study tested four different training thresholds, 0.1, 0.25, 0.4 and 0.55, at which time adjustment of the network weights stopped. The "best" model for each hexagon was chosen as the one with the lowest sum of the square error on the test data set (1990,1991). This model was then used in all subsequent analysis.

MODEL EVALUATION

The ANN predictions for the years 1982 through 1990 were generated by iterating the model forward one year using actual data from the year before as inputs. For example, actual 1981 data was used to produce predictions for 1982 and so forth. The 1991 ANN prediction is a two iteration case in which the model predictions for 1990 were used as inputs to produce the prediction. The linear trend model is plotted as the regression line representing the trend of the data between 1981 and 1989 extended through the test set years of 1990 and 1991. For each hexagon, the "best" model was chosen as the one which had the lowest sum of the square error on the test data set (fig. 5). In most cases the ANN model was able to lock into a pattern of development in the training data and produce predictions which were superior to the linear trend model. The ANN models with small learning criteria (< 0.1) approximate the trend of training set data quite well, while those models with larger learning criteria do not represent the training data as well.

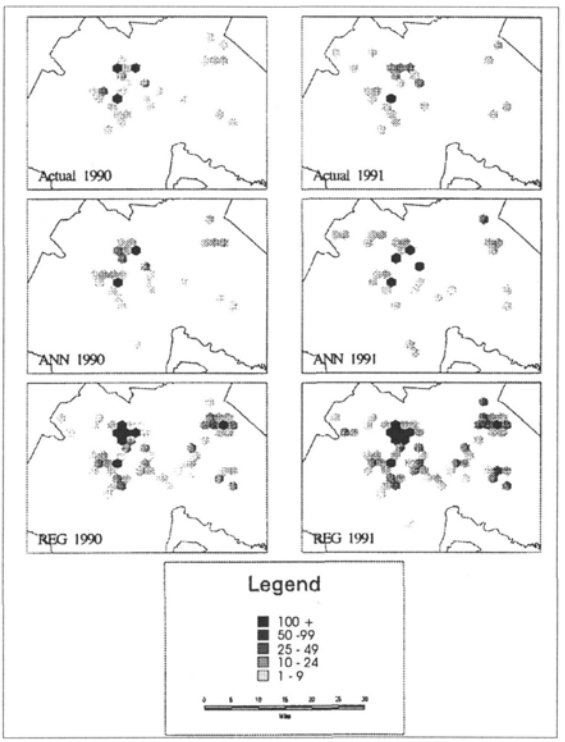


Fig. 5 Predicted building permits versus actual

A few cases illustrate how some of the models predicted the building permit data (fig. 6). In case 1 an ANN model was able to accurately approximate the nonlinear trend of the building permit data, including the test data set. Case 2 is an interesting case in which the trend of the building permit data does not match the hypothetical time series. The ANN model was able to pick up on the appropriate pattern and predict the increase in permits occurring in 1991. The prediction is not drastically different from the linear trend model in this year, but the ability of the ANN model to adjust for this upswing is evident. Case 3 is justification for using higher learning criteria during the training phase of the model development. The increased learning criteria allowed the ANN model to ignore what may be noise in the training set data and still have the ability to model the general trend of the data and produce desirable results on the test set data. This property gives ANN's a distinct advantage over linear trends when modeling dynamic systems. Case 4 illustrates a problem involved with modeling dynamic systems. In this case the model has picked up on an inappropriate trend in the data and has projected the growth upwards in 1991 when in actuality it is tending towards zero.

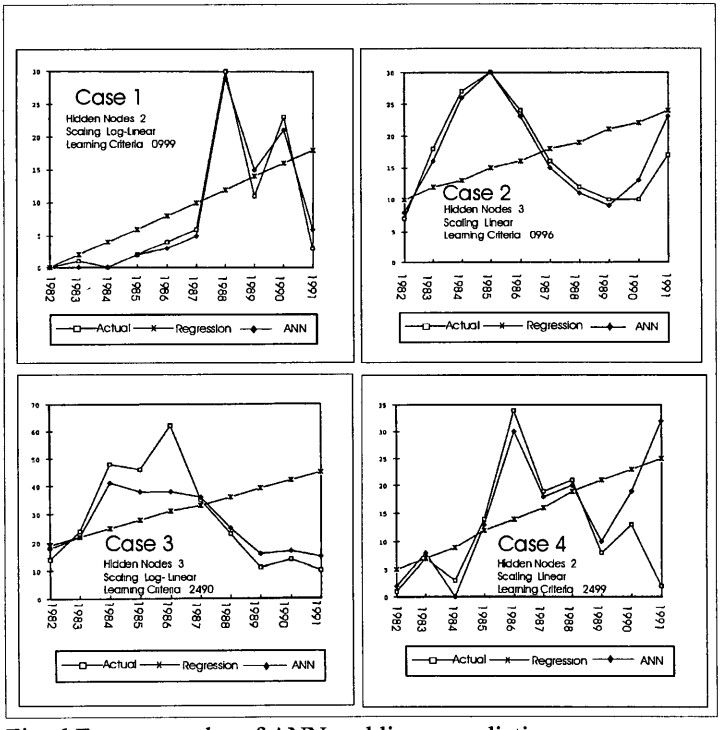


Fig. 6 Four examples of ANN and linear predictions

The final evaluation was based on regressing each model's (ANN and regression) predictions against the actual building activity occurring within each

hexagon in both 1990 and 1991 (fig. 7). This resulted in four bivariate regression equations . Over the 94 hexagons in the study area, the ANN models were shown to be superior to the linear trend models in that the ANN models produced predictions which were closer to the actual data values than did the linear trend model. For the 1990 ANN one iteration model the regression parameters for the one iteration ANN model predictions for 1990 the r^2 was 0.83. In contrast the regression model had an r^2 of 0.61. The intercept for the regression estimate was 9.49 which was significantly different than zero. These parameters indicate that the linear trend predictions for 1990 consistently over-predicting the number of building permits throughout the study area.

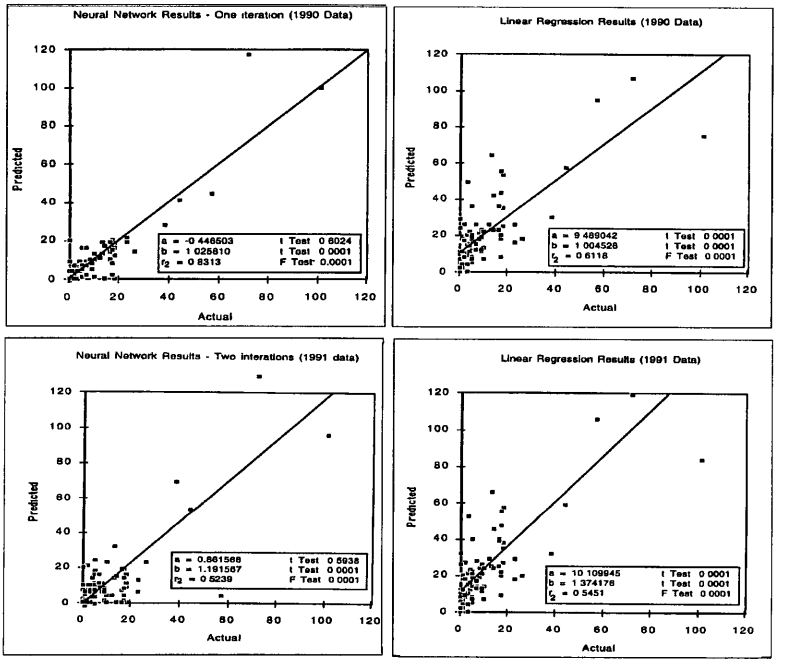


Fig. 7 Regression results of ANN versus linear model.

The regression model for the two iteration ANN model predictions for 1991 and the actual number of permits had an r^2 of only 0.52 and a slope of 1.19. The expected growth of the error term between the trajectory of the ANN model predictions and the actual data values is apparent in these results. The fact that the two iteration case had an intercept of zero and a slope near 1 which indicate that while the models do not perform consistently for all hexagons, they do produce results which are around the desired values. The comparison of the linear trend predictions for 1991 and the actual number of permits generated an r^2 of 0.55 with

a slope of 1.37. As was the case with the results of the linear trend on the 1990 test data, the linear predictions for 1991 are consistently higher than the actual number of permits. The difference between the two approaches is most clearly demonstrated by the fact that the intercept of the ANN model is approximately zero (0.86) and the intercept of the linear trend model significantly greater than zero (10.11).

CONCLUSIONS

A spatial temporal database was constructed by aggregating a database of building permit data to a tessellation of regular hexagons. Artificial neural network techniques were used to develop models that replicated the time series for each hexagon. These models were evaluated by comparing the predictions of the models for two years of data the model did not see during the training phase of model development against the predictions from a linear trend model. The results of the study provide strong evidence for power of an ANN to model non-linear trends. For the one iteration case, the ANN model was able to produce predictions over the entire study area which closely resembled the actual values, while the linear trend model produced results which consistently overestimated the actual number of permits. The models were able to adjust to the variations in the building permit data without being aware of fluctuations in the local economy, available land for development, accessibility, etc. The success of the approach used here is encouraging for modeling systems, such as urban dynamics, for which the relationships between the underlying mechanisms are not well understood and for which precise data is not available.

BIBLIOGRAPHY

Alexander, C. (1965) "A City is Not a Tree", *Architectural Forum* **122** (1), 58-61 and (2), 58-62.

Batty, M, and Xie, Y. (1994) "Urban Analysis in a GIS Environment" from *Spatial Analysis and GIS*, edited by Fotheringham, S. and Rogerson, P., Taylor & Francis, Bristol, PA, pp. 189-219.

Chakraborty, K., Mehrotra, K., Mohan, C.K. and Ranka, S. (1992) "Forecasting the Behavior of Multivariate Time Series Using Neural Networks", *Neural Networks*, vol. 5, no. 6, pp. 961-970.

Chapin F.C. and Weiss S.F. (1968) "A Probabilistic Model for Residential Growth", *Transportation Research* vol. 2, pp. 375-390.

Fletcher, D. and Goss, E. (1993) Forecasting with neural networks: An application using bankruptcy data, Information and Management, 24, Elsevier .

Halls, J.N., Cowen, D.J. and Jensen, J.R. (1994) "Predictive Spatio-Temporal Modeling in GIS", *Proceedings of the Sixth International Symposium on Spatial Data Handling*, vol. 1, pp. 431-448.

Heermann, P.D. and Khazenie, N. (1992) Classification of Multispectral Remote Sensing Data Using a Back-Propagation Neural Network, IEEE Transactions on Geoscience and Remote Sensing, Vol. 30, No. 1, January 1992, IEEE.

Lodewyck, R.W. and Deng, P, (1993) Experimentation with a back-propagation neural network: An application to planning end user system development, Information and Management, 24, Elsevier Science Publishers.

Lowe, D. and Webb, A.R. (1991) "Time series prediction by adaptive networks: a dynamical systems perspective", *IEEE Proceedings-F*, vol. 128, no. 1, February 1991, pp. 17-24.

Vemuri, V.R.. and Rogers, R.D.. (1994) *Artificial Neural Networks: Forecasting Time Series*, IEEE Computer Society Press, Los Alamitos, CA.

AN EVALUATION OF CLASSIFICATION SCHEMES BASED ON THE STATISTICAL VERSUS THE SPATIAL STRUCTURE PROPERTIES OF GEOGRAPHIC DISTRIBUTIONS IN CHOROPLETH MAPPING

Robert G. Cromley and Richard D. Mrozinski

Department of Geography
University of Connecticut
Storrs, CT 06268-2148
USA

ABSTRACT

In choropleth mapping, most classification schemes that have been proposed are based on the properties of the data's statistical distribution without regard for the data's spatial distribution. However, one of the more important tasks associated with choropleth map reading is the task of regionalization and identifying spatial patterns. For this reason some authors have proposed class interval selection procedures that also consider spatial contiguity. This paper evaluates different classification schemes based on a data set's statistical as well as its spatial distribution. A comparison of the Jenks' optimal classification that minimizes within group variation and a contiguity based method that minimizes boundary error show that the latter method was not as strongly influenced by changes in the statistical distribution and produced a more complex map as measured by the number of external class boundaries present in the map display. (Keywords: data classification, choropleth mapping, spatial autocorrelation)

INTRODUCTION

Numerous classification methods for choropleth mapping have been proposed and evaluated (see Jenks and Coulson, 1963; Evans, 1977; Cromley, 1996). In general, most traditional and even optimal classification schemes such as the Jenks' optimal classification (Jenks, 1977) that minimizes total within group variation are based on the properties of the data's statistical distribution without regard for the data's spatial distribution. However, the task of regionalization is one of the more important tasks associated with choropleth map reading. Several authors (Monmonier, 1972; Cromley, 1996) have proposed class interval selection procedures that also consider spatial contiguity. The purpose of this paper is to evaluate classification schemes based on a data set's statistical distribution versus its spatial distribution. For this evaluation, the Jenks' optimal classification was chosen to represent schemes based on

statistical properties and Cromley's boundary error method (Cromley, 1996) was chosen to represent schemes that incorporate the spatial contiguity of the data values.

BACKGROUND

It has long been recognized that classification schemes have a major impact on the visualization of choropleth maps. Because the classification process transforms interval or ratio data into ordinal classes, information is lost converting individual algebraic numbers into ordinal classes. Secondly, grouping N unique data values into p different classes (N>p) implies that there are (N-1)!/(N-p)!(p-1)! different classification groupings. Monmonier (1991) has demonstrated how easy it is to distort the visual pattern of the data by manipulating the class interval breaks. The ambiguity caused by classification prompted Tobler (1973) to propose classless maps as an alternative to the classed choropleth map in which algebraic numbers are directly converted into graphic values. An areal table map (see Jenks, 1976) reduces this ambiguity even more by displaying the algebraic numbers directly within the outline of each area but visually recognizing patterns of spatial autocorrelation in geographic data sets would be more difficult.

To ensure that classification schemes try to represent the data distributions, different schemes have been evaluated within respect to how much error is associated with the classification (Jenks and Caspall, 1971) and the impact of class interval systems also have been analyzed with respect to the evaluation of pattern relationships (Monmonier, 1972; Olson, 1975; Dykes, 1994; Cromley and Cromley, 1996). While there are problems associated with any classification, well constructed classifications can aid the reader in most mapping tasks. Mak and Coulson (1991) found in perception tests that classed choropleth maps using the Jenks' optimal classification system (Jenks, 1977) were significantly better than classless maps for the task of value estimation although there was no significant difference in regionalization tasks.

The problem addressed here is to examine visually and quantitatively how well different classification schemes preserve the underlying spatial structure of the data. Cromley and Cromley (1996) found that quantile schemes frequently used in map atlases represented spatial patterns worse than classification based on minimizing the error associated with class boundaries. However, quantile classifications also generally produce worse representations than other classifications with respect to most statistical properties. The comparison here will be made between the Jenks' optimal classification and the boundary error method formulated by Cromley (1996).

Both of these methods are "optimal" in the sense that each minimizes or maximizes some performance measure. Both classification schemes are derived

from the same basic model. Based on Monmonier's work (1973) applying location-allocation models to the classification problem, Cromley (1996) has shown that optimal classification can be solved as a shortest path problem over an acyclic network. Using the number line associated with the sorted data values as an acyclic network, each arc connecting two nodes in the network represents a class interval containing data values. Given n points in the original data set, there would be n+1 nodes and n(n+1)/2 arcs in the acyclic network. For classifying data values in choropleth mapping, the cost value associated with each arc corresponds to an objective performance measure. By varying the definition of this performance measure, alternative optimal classifications can be constructed (Cromley, 1996).

Within the framework of this generic optimal classification model, the Jenks' optimal classification scheme defines the cost value for each class as the within class variation. By minimizing this value over all groups, the classification minimizes the total within group variation so that as much of the overall variation is "explained" by the classification as much as possible. The Jenks' optimal classification is also referred to as the VGROUP classification for the remainder of this paper.

Boundary error occurs whenever the boundaries between the classed areas on a map, referred to as external class boundaries, do not align with the major breaks in a three dimensional representation of the statistical surface (Jenks and Caspall, 1971). Classification should result in the boundaries lying within a group of contiguous area units, referred to as internal class boundaries, corresponding to minor breaks in the surface while the boundaries separating a grouping correspond to the major breaks in the surface. Within the generic model, the cost value for each class is now defined as the variation between the right- and left-hand area units associated with each internal class boundary. Only the deviations associated with boundaries separating area units within a class are counted while the deviations associated with boundaries separating area units belonging to different classes are not counted. By minimizing this cost value over all classes, the internal class boundaries should correspond to minor breaks in the surface and any regionalization should be fairly homogeneous. Because this classification (referred to as BGROUP) utilizes information regarding the relative location of data values, its implementation requires a topological data structure for the base map as well as the data values themselves.

DATA

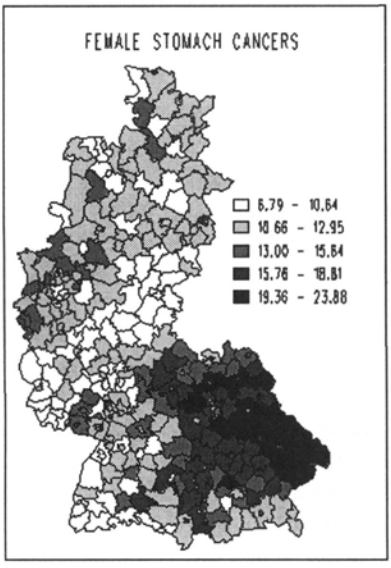
For evaluating these different approaches to classification, a cancer mortality data set was selected from West Germany originally published and mapped in *Atlas of Cancer Mortality in the Federal Republic of Germany* (Becker *et al.*, 1984). The data in this atlas were collected at the level of the **kreise**

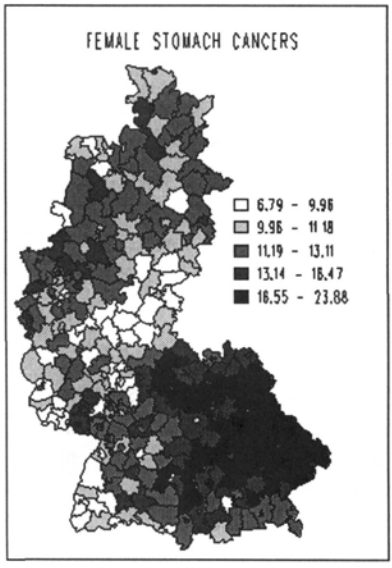
administrative unit for which mortality rates were estimated by the authors. Overall, there were 328 observations for each cancer; the **kreise** of West Berlin was removed from the original data because it was a detached unit and did not share common boundary with any other unit. Female stomach cancer, which was highly positively autocorrelated in West Germany, and ovarian cancer, which was randomly distributed over space, were chosen to test the effect of spatial arrangement on each classification. No negatively autocorrelated patterns were used because these patterns rarely occur in most geographic processes. Each of these cancer data sets also were slightly positively skewed in their respective statistical distributions.

In addition to mapping each cancer by both the Jenks' optimal classification scheme, VGROUP, and the spatial structure method, BGROUP (see Figures 1 and 2), three artificial data distributions were classified mapped for each cancer. These artificial data distributions are created to add differing levels of skewness in the statistical distribution for the same basic spatial arrangement of data values. A linear, arithmetic, and geometric progression (see Jenks and Coulson, 1963) of data values were generated and then assigned to **kreise** such that the ordinal position of each **kreise** was the same for each progression as for female stomach cancer and then the ordinal position of each **kreise** was the same for each progression as for ovarian cancer. Thus, each progression has the same statistical distribution for each cancer but a different spatial arrangement. Finally, to keep the number of maps to a manageable number, only a five class map was produced for each original cancer and every progression/spatial arrangement combination.

RESULTS

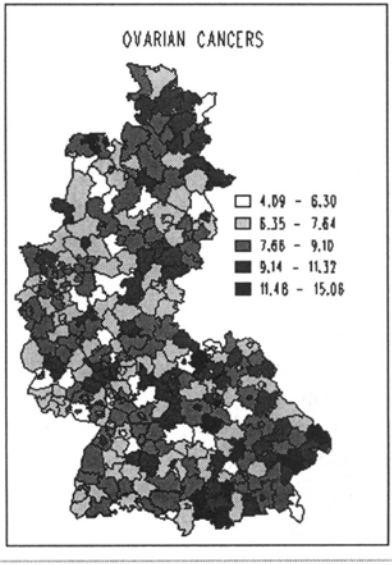
The Jenks' optimal classification of original female stomach cancer data was somewhat different than that for ovarian cancers as these two data sets had different statistical distributions although both were positively skewed (see Table 1). However, because the Jenk's optimal classification is based solely on the statistical distribution, the class intervals for the three progressions were the exactly the same for each progression regardless of how the values were spatially arranged. Secondly, the linear progression resulted in the same number of observations in each class. In this one case, optimal classification generates the same result as traditional quantile or equal interval schemes. Thirdly, as each artificial distribution became more positively skewed, more observations were grouped into the lower classes because the Jenks' classification is influenced by extreme values.

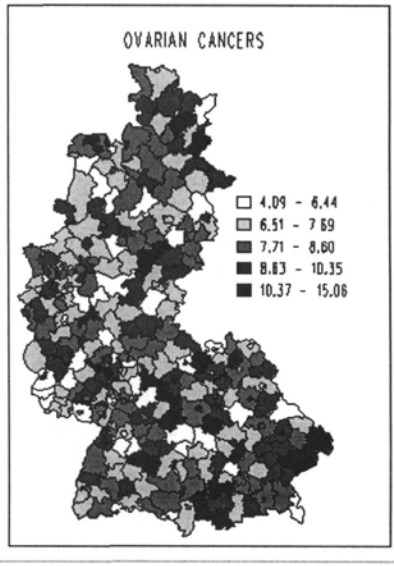




(a) VGROUP (b) BGROUP

Figure 1: Classified Female Stomach Cancers.





(a) VGROUP (b) BGROUP

Figure 2: Classified Ovarian Cancers.

The BGROUP classification method, in contrast, always produced a different set of class intervals for each statistical distribution/spatial arrangement combination (see Table 1). The number of observations for the linear progression that was positively autocorrelated (matched with the female stomach cancer arrangement) had fewer observations in the extreme classes than for the linear progression that was randomly arranged. As each progression became more positively skewed, more observations were grouped into the lowest data class although at a much lower rate than in the Jenks' optimal method.

TABLE 1
Number of Observations in each Class

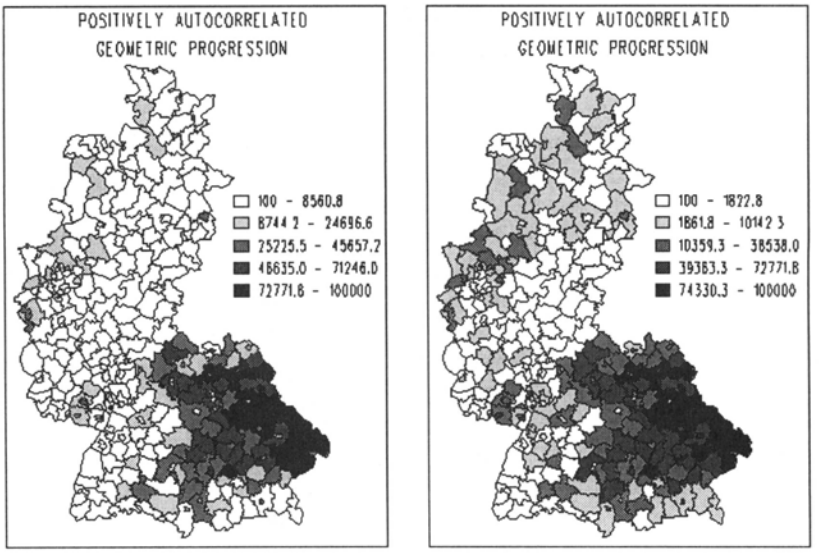
		Positively Autocorrelated		Random	
		VGROUP	BGROUP	VGROUP	BGROUP
Original Class Data*	#1	84	49	33	44
	#2	132	72	104	96
	#3	56	98	120	84
	#4	36	63	54	71
	#5	19	45	16	32
Linear Class Progression	#1	66	49	66	82
	#2	65	72	65	70
	#3	65	84	65	62
	#4	65	76	65	49
	#5	66	46	66	64
Arithmetic Class Progression	#1	119	103	119	84
	#2	67	59	67	71
	#3	53	57	53	66
	#4	46	63	46	63
	#5	42	45	42	43
Geometric Class Progression	#1	211	138	211	145
	#2	50	81	50	69
	#3	29	63	29	49
	#4	21	30	21	36
	#5	16	15	16	28

*The original data for the positively autocorrelated distribution were Female Cancers and the original data for the random distribution were Ovarian Cancers.

The overall result was that the spatial structure classification retained a higher level of visual complexity as the data distributions became more positively skewed especially for the data that was more positively autocorrelated (see Figures 3 and 4). In general, the visual complexity of a map increases as the spatial autocorrelation moves from high positive autocorrelation to random to high negative autocorrelation (Olson, 1975). The VGROUP classification of the geometric progression displayed a much larger homogeneous region of low values for the positively correlated distribution than for the spatially random distribution (see Figures 3a and 4a). Because the number of observations in each class was more balanced for the BGROUP classification than the VGROUP classification, higher level of visual complexity was retained for data set. For example, the large white area associated with the lowest class of Figure 3a is broken up into other classes in Figure 3b especially in the northern tier of **kreise**.

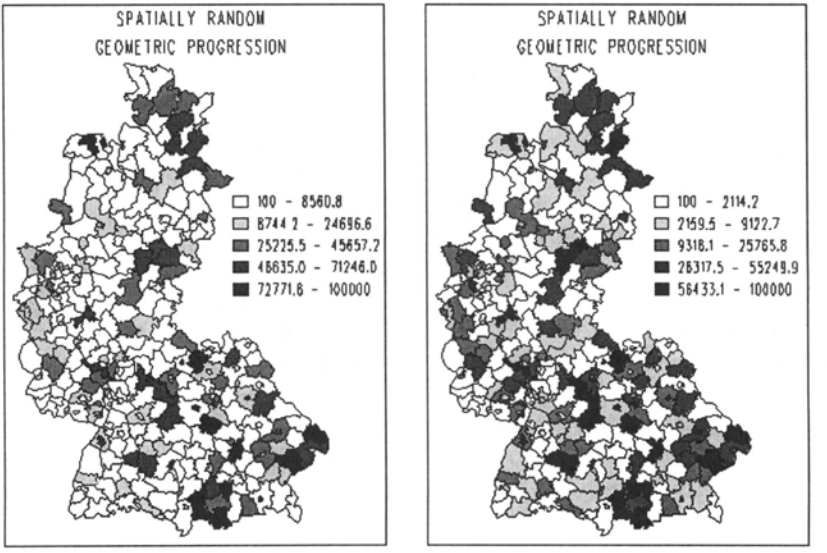
Quantitatively, this is measured first by the number of external and internal class boundaries generated by each classification. Because the boundaries between classes dominate the visual representation (Jenks and Caspall, 1971), the more external boundaries, the more visually complex the representation. In Table 2, the number of external and internal boundaries are matched against the Moran I coefficient for each data set. Regardless of the level of autocorrelation, the BGROUP classification always retained more external class boundaries than the VGROUP classification. Secondly, the BGROUP classification retained a similar number of external boundaries over the different data progressions.

Another measure of spatial autocorrelation and map complexity that has been used for map classifications is Kendall's tau (Monmonier, 1974; Olson, 1975). Similar to Moran's I index for metric data, Kendall's tau ranges from +1.0 to -1.0 for ordinal data with +1.0 associated with perfectly positive autocorrelation. With respect to Kendall's tau, the results were more mixed; for the positively autocorrelated distributions, the Kendall's tau value associated with BGROUP classification was higher for the stomach cancers data and the geometric progression and lower for the linear and arithmetic progressions. For the spatially random distributions, the Kendall's tau value was lower for the three progression and slightly higher for the ovarian cancers data. In general, the tau values were about the same for both classification with the exception of the positively autocorrelated geometric progression. Kendall's tau is influenced by an uneven number of observations in each grouping; the more uneven, the lower the value will be. Because the increasing skewness in the data resulted in the VGROUP's highly uneven number of observations in each class, the tau value is decreased.



(a) VGROUP (b) BGROUP

Figure 3. Classified Positively Autocorrelated Geometric Progression.



(a) VGROUP (b) BGROUP

Figure 4. Classified Spatially Random Geometric Progression.

TABLE 2
A Comparison of the Level of Spatial Autocorrelation
By Different Measures of Complexity

		Original Data*	Linear Progression	Arithmetic Progression	Geometric Progression
Positively Autocorrelated					
Moran I		0.759	0.636	0.723	0.807
# External Boundaries	VGROUP	466	515	483	322
	BGROUP	548	547	530	467
Kendall's Tau	VGROUP	0.461	0.485	0.493	0.381
	BGROUP	0.482	0.477	0.472	0.458
Random					
Moran I		0.080	0.053	0.069	0.095
# External Boundaries	VGROUP	600	664	599	426
	BGROUP	666	681	679	599
Kendall's Tau	VGROUP	0.036	0.036	0.050	0.036
	BGROUP	0.039	0.022	0.020	0.025

*The original data for the positively autocorrelated distribution were Female Cancers and the original data for the random distribution were Ovarian Cancers.

CONCLUSIONS

As expected, the Jenks' optimal classification was more strongly influenced by changes in the statistical properties of a data distribution than the classification that minimized boundary error. In all cases, the BGROUP classification resulted in a map display that had more external class boundaries than the Jenks' optimal classification. However, Kendall's tau measure for computing spatial autocorrelation for grouped ordinal data did not detect much difference between the two classifications except for the positively autocorrelated geometric progression. The overall result is that the BGROUP classification scheme probably retains more visual complexity and more homogeneous regions than the Jenks' optimal classification scheme.

REFERENCES

Becker, N., R. Frentzel-Beyme, and G. Wagner. (1984). *Atlas of Cancer Mortality in the Federal Republic of Germany*, Berlin: Springer-Verlag.

Cromley, R. (1996). A comparison of optimal classification strategies for choroplethic displays of spatially aggregated data. *International Journal of Geographical Information Systems*, 10:404-424.

Cromley, E. and R. Cromley. (1996). An analysis of alternative classification schemes for medical atlas mapping. *European Journal of Cancer*, 32A:1551-1559.

Dykes, J. (1994). Visualizing spatial association in area-value data. Chapter 11 in *Innovations in GIS I*, M.F. Worboys (ed.), London: Taylor and Francis, 149-159.

Evans, I.S. (1977). The selection of class intervals. *Transactions of the Institute of British Geographers*, 2:98-124.

Jenks, G. (1976). Contemporary statistical maps -- Evidence of spatial and graphic ignorance. *The American Cartographer*, 3:11-19.

Jenks, G. (1977). *Optimal Data Classification for Choropleth Maps*. Occasional paper No. 2, department of geography, University of Kansas.

Jenks, G. and F.C. Caspall. (1971). Error on choroplethic maps: Definition, measurement, reduction. *Annals of the Association of American Geographers*, 61:217-244.

Jenks, G. and M. Coulson. (1963). Class Intervals for Statistical Maps. *International Yearbook of Cartography*, 3:119-134.

Mak, K. and M. Coulson. (1991). Map-user response to computer-generated choropleth maps: Comparative experiments in classification and symbolization. *Cartography and Geographic Information Systems*, 18:109-124.

Monmonier, M. (1972). Contiguity-biased class-interval selection: A method for simplifying patterns on statistical maps. *Geographical Review*, 62:203-228.

Monmonier, M. (1973). Analogs between class-interval selection and location-allocation models. *The Canadian Cartographer*, 10:123-131.

Monmonier, M. (1974). Measures of pattern complexity for choroplethic maps. *The American Cartographer*, 1:159-169.

Monmonier, M. (1991). *How to Lie with Maps*. Chicago: University of Chicago Press.

Olson, J. (1975). Autocorrelation and visual map complexity. *Annals of the Association of American Geographers*, 65:189-204.

Tobler, W. (1973). Choropleth maps without class intervals? *Geographical Analysis*, 5:262-265.

MAPPING MULTIVARIATE SPATIAL RELATIONSHIPS FROM REGRESSION TREES BY PARTITIONS OF COLOR VISUAL VARIABLES

Denis White and Jean C Sifneos
Department of Geosciences
Oregon State University
Corvallis, OR, 97331 USA

ABSTRACT. In classification and regression tree (CART) analysis, the observations are successively partitioned into a prediction tree. At each node in the tree, the CART algorithm searches for the value of one of the predictor variables that explains the greatest amount of variation in the response variable. The observations are split into two groups at each node according to this splitting criterion until the tree reaches a size that balances predictive power and parsimony. We illustrate a method for mapping the spatial relationships in a prediction tree when the cases are spatial. Each leaf in the tree has a unique set of predictor variables and corresponding value ranges that predict the value of the response variable at the observations belonging to the leaf. If the tree is arranged such that observations with lower values of the splitting variable are always on the left at each node, then there is an unambiguous ordering to the tree. One method for assigning mapping symbols to the observations of the leaves is by locating each leaf in a corresponding position along the continuum of one of the color visual variables. Observations that are closer in perceptual value to others indicate a closer relationship in the structure of prediction.

INTRODUCTION

Classification and regression trees (CART) are a multivariate analysis technique made popular by Breiman *et al.* (1984). Applications are varied: examples include machine learning (Crawford, 1989), medicine (Efron and Tibshirani, 1991), optical character recognition (Chou, 1991), soil classification (Dymond and Luckman, 1994), forest classification (Moore *et al.*, 1991), vegetation ecology (Davis *et al.*, 1990; Michaelson *et al.*, 1994), animal distribution (Walker, 1990), biodiversity (O'Connor *et al.*, 1996), and others.

In an application where the cases are spatial locations, the geography of the prediction tree results may reveal insights into mechanistic relationships between the predictors and the response. Mapping residuals from the prediction tree may also help to identify missing variables or gaps in knowledge. Previous work in mapping CART results includes Davis *et al.* (1990), Walker (1990), Moore *et al.* (1991) and O'Connor *et al.* (1996). We explore this idea by proposing an objective method for assigning map symbols to the leaves of regression (or classification) trees. We illustrate this mapping method with both simulated and real data.

METHODS

In regression tree development, the midpoints between all values of all of the predictor variables that are present in the data form the possible splits for the tree. In the first step, sums of squares of differences between the observations and their means are computed for all binary divisions of the observations formed by all of the splits. The minimum sum determines the split. The observations are then divided into two sets based on the split and the process recursively repeats on the two descendent sets. Splitting continues until a stopping criterion is reached. We used the cross-validation pruning techniques of Breiman *et al.* (1984), as implemented by Clark and Pregibon (1992), and as investigated by Sifneos *et al.* (in preparation), to determine the optimal size of trees.

We prepared two simulated data sets as examples. The first set consisted of three predictor variables defined as two level (x1), or three level (x2 and x3), step functions. The response variable (y) was defined as a four level step function. All variables were defined on a 10 by 10 grid, simulating a spatial surface. The steps were defined on one quarter or one half of the grid (Figure 1). The second simulated set also consisted of three predictors, but these were samples from a lognormal distribution (x1) and two different normal distributions (x2 and x3), respectively. The response (y) was defined differently in each quadrant of the grid to simulate the contingent effects of hierarchical interactions that CART is well suited to analyze. The first quadrant was defined as $y = x1 + 2x2 + 3x3$, for example, and the other quadrants as indicated in Figure 2.

In addition, we used portions of a data set from a fish biodiversity study (Rathert *et al.*, in preparation). For illustrating the regression tree mapping method we used total fish species richness, including native and introduced species, as a response variable. We used 20 predictor variables representing climatic, elevational, hydrographic extent, and human impact effects (Figure 4). All variables were provided for 375 equal area sample units covering the state of Oregon. (The variable representing the length of 4th and higher order streams in each sampling unit is not shown in Figure 4.)

We can think of the mapping of regression trees in the framework of measurement scales. The terminal nodes, or leaves, of a tree contain observations that have a unique chain of prediction rules with respect to other leaves. The uniqueness property confers at least a nominal scaling on the leaves. Because the predictor splits can be arranged in an unambiguous order with lower values of continuous variables, for example, always appearing in the left branches, an ordinal scaling can be imposed as well. (Nominal predictor variables can meet this criterion with an arbitrary ordering of categories.) More ambitiously, we may attempt to convey distance in prediction space by mapping leaf positions to an interval scale. Color visual variables are good candidates to symbolize these scaling distinctions. In this paper we present tree mapping using

ordinal scaling of the value or brightness dimension. We select a number, equal to the number of leaves, of equally-spaced division points along the value scale. In another paper we present a more refined implementation of this idea using a recursive partition of the hue spectrum to mimic the recursive partitioning of the observation space by the regression tree (White and Sifneos, in preparation).

RESULTS

The stepped prediction and response surfaces (Figure 1) produced a simple tree that has perfect prediction. That is, the variation explained, computed as the ratio of sums of squares in the leaves to that of the root node, subtracted from one, is exactly equal to 1. The tree diagram expressing the prediction relationship (Figure 1) followed the pattern of predictors precisely: the first split recognized the division of the observation grid into two halves by x1; the left branch of the tree representing the top half of the grid was split by x2; and the right branch representing the bottom by x3. A multiple linear regression on this data also achieved perfect prediction with an R-Squared of 1.

The contingent response from normally and lognormally distributed predictors (Figure 2) produced, in one realization, a tree with six leaves (Figure 3). We applied the value scaling to the leaves and mapped the prediction groups of observations on the simulated study area grid (Figure 3). The variation explained by the tree was 0.71. A multiple regression with no interactions between predictors produced a R-Squared of 0.28. (A multiple regression including interactions between predictors would have a higher R-Squared.)

A regression tree analysis of the fish data set produced a tree with seven leaves (Figure 5). Each of the six splits used a different predictor variable. The variation explained by the tree was 0.72. A multiple regression fit with no interactions had a R-Squared of 0.50, using seven predictor variables determined through stepwise procedures. The map of prediction groups from the regression tree revealed a strong east-west structure in Oregon (Figure 6). On the west side of the Cascades, climate and elevation variables formed the prediction, while on the east side, stream length variables. The value scale mapping of leaf prediction groups with gray tones helped to identify this structure. Comprehensive analysis of this data and an interpretation of the biogeography will be found in Rathert *et al.* (in preparation).

ACKNOWLEDGEMENTS

We acknowledge support from agreements CR 821672 between US EPA and Oregon State University, PNW 92-0283 between US Forest Service and OSU, DW12935631 between US EPA and USFS, and DOD SERDP Project #241-EPA. This research has not been officially reviewed by US EPA and no endorsement should be inferred.

REFERENCES

Breiman, L, J H Friedman, R A Olshen, C J Stone. (1984). *Classification and regression trees.* Chapman & Hall, New York.

Chou, P A. (1991). Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:340-354.

Clark, L A, D Pregibon. (1992). Tree-based models. In: *Statistical models in S.* JM Chambers & TJ Hastie, editors. Wadsworth & Brooks, Pacific Grove, CA. pp. 377-419.

Davis, F W, J Michaelsen, R Dubayah, J Dozier. (1990). Optimal terrain stratification for integrating ground data from FIFE. In: *Symposium on FIFE, First ISLSCP Field Experiment.* Amer. Meteor. Soc., Boston, MA. pp. 11-15.

Dymond, J R, P G Luckman. (1994). Direct induction of compact rule-based classifiers for resource mapping. *Int. J. Geog. Info. Sys.*, 8:357-367.

Efron, B, R Tibshirani. (1991). Statistical data analysis in the computer age. *Science*, 253:390-395.

Michaelsen, J, D S Schimel, M A Friedl, F W Davis, R C Dubayah. (1994). Regression tree analysis of satellite and terrain data to guide vegetation sampling and surveys. *J. Veg. Sci.*, 5:673-686.

Moore, D M, B G Lees, S M Davey. (1991). A new method for predicting vegetation distributions using decision tree analysis in a geographic information system. *Environmental Management*, 15:59-71.

O'Connor, R J, M T Jones, D White, C Hunsaker, T Loveland, B Jones, E Preston. (1996). Spatial partitioning of environmental correlates of avian biodiversity in the conterminous United States. *Biodiversity Letters,* in press.

Rathert, D, D White, J C Sifneos, R M Hughes. (In preparation). Environmental correlates of species richness in Oregon freshwater fishes.

Sifneos, J C, D White, N S Urquhart, D Schafer. (In preparation). Selecting the size of regression tree models.

Walker, P A. (1990). Modelling wildlife distributions using a geographic information system: kangaroos in relation to climate. *J. Biogeog.*, 17:279-289.

White, D, J C Sifneos. (In preparation). Mapping multivariate spatial relationships from regression trees by recursive binary partitions of the spectrum.

Figure 1

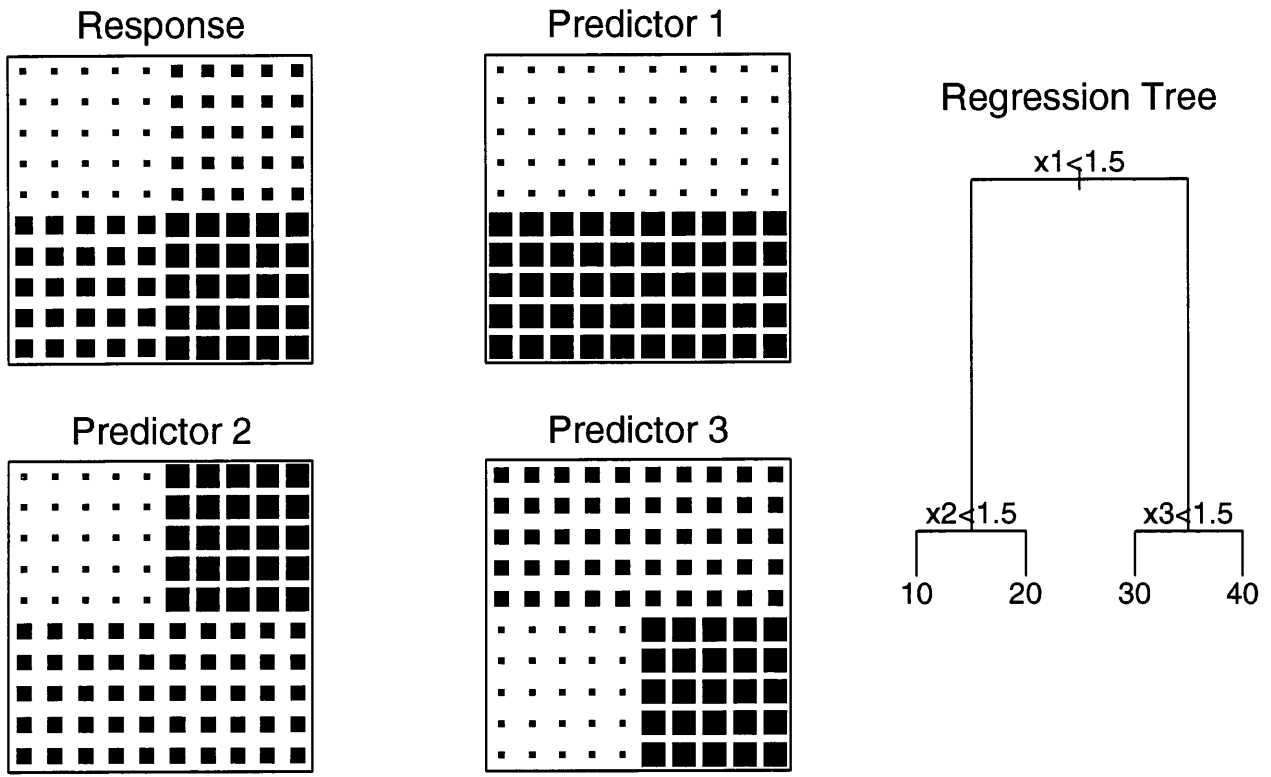

Stepped Response and Predictors
Response
Predictor 1
Predictor 2
Predictor 3
Regression Tree
x1<1.5
x2<1.5
x3<1.5
10
20
30
40

Figure 2

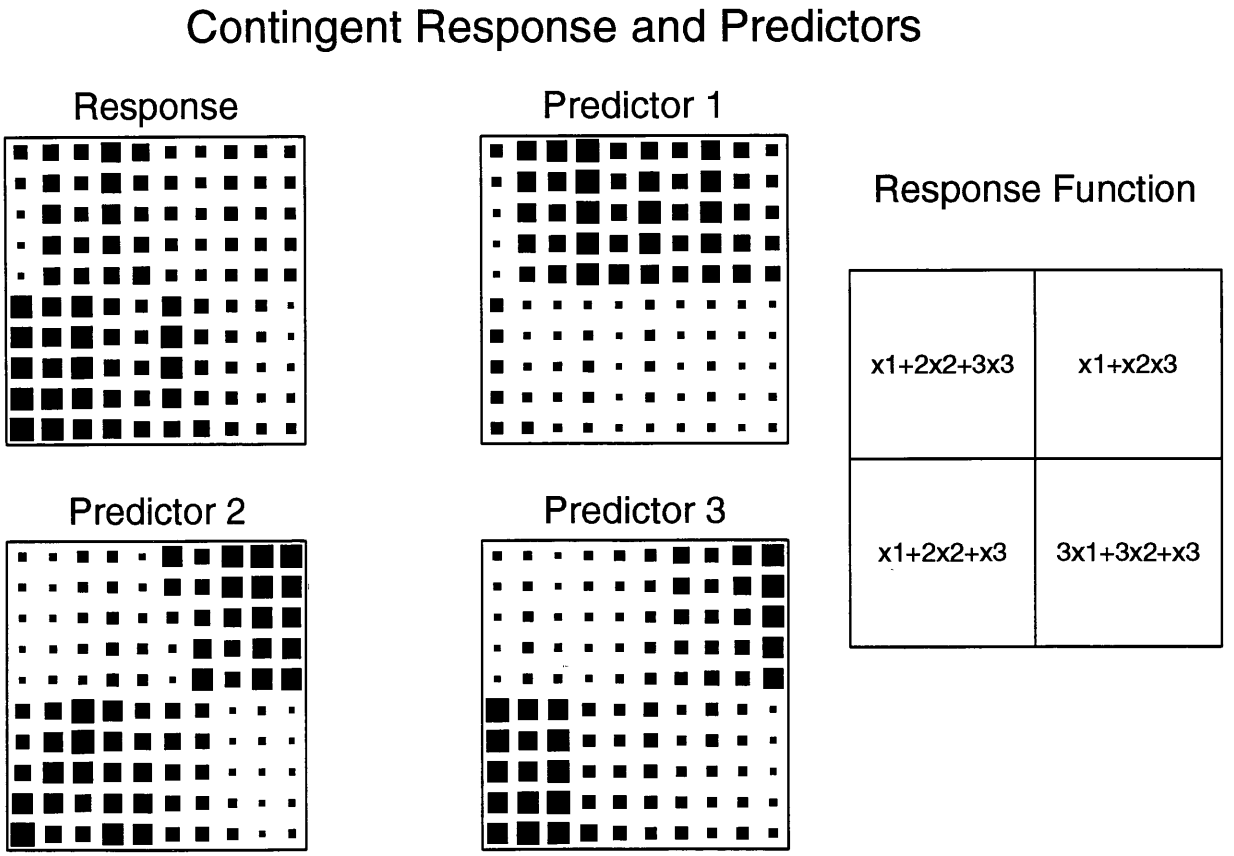
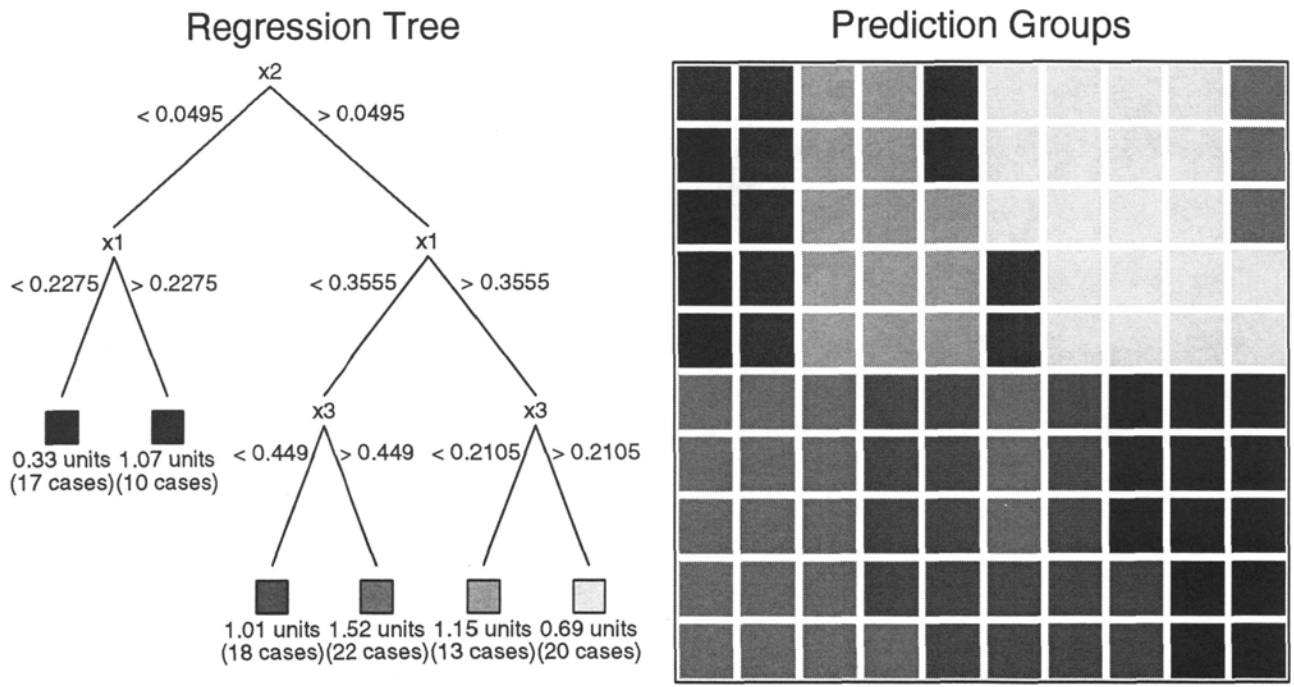

Contingent Response and Predictors
Response
Predictor 1
Predictor 2
Predictor 3
Response Function
x1+2x2+3x3
x1+x2x3
x1+2x2+x3
3x1+3x2+x3

Figure 3


Contingent Response
Regression Tree
Prediction Groups
x2
< 0.0495
> 0.0495
x1
x1
< 0.2275
> 0.2275
< 0.3555
> 0.3555
x3
x3
0.33 units
(17 cases)
1.07 units
(10 cases)
< 0.449
> 0.449
< 0.2105
> 0.2105
1.01 units
(18 cases)
1.52 units
(22 cases)
1.15 units
(13 cases)
0.69 units
(20 cases)

Fish Species Richness and Predictors
Fish.Species
Jan.Temp.Min
Jul.Temp.Max
Annual.Temp.Range
Ann.Temp.Rng.Std.Dev
Monthly.Temp.Std.Dev
Annual.Precip.Min
Annual.Precip.Max
Annual.Precip.Range
Annual.Precip.Mean
Elevation.Min
Elevation.Max
Elevation.Range
First.Order.Streams
Second.Order.Streams
Third.Order.Streams
Intermittent.Streams
Perennial.Streams
Road.Length
Human.Population

Figure 4

Fish Species Richness
Regression Tree
Annual.Temp.Range
< 16.625
> 16.625
Elevation.Range
< 484.022
> 484.022
Perennial.Streams
< 45.33
> 45.33
Jan.Temp.Min
< -7.5
> -7.5
Higher.Order.Streams
< 35.785
> 35.785
Annual.Precip.Range
< 953
> 953
29 species
(16 cases)
10 species
(20 cases)
2 species
(69 cases)
19 species
(24 cases)
6 species
(105 cases)
14 species
(43 cases)
9 species
(98 cases)

Figure 5

Fish Species Richness

Prediction Groups

Figure 6

HOW MANY REGIONS? TOWARD A DEFINITION OF REGIONALIZATION EFFICIENCY

Ferko Csillag[o] and Sandor Kabos[oo]
[o] Department of Geography, University of Toronto, Erindale College, Mississauga, ONT, L5L 1C6, Canada | <fcs@eratos.erin.utoronto.ca>
[oo] Mathematical Statistical Group, Institute of Sociology, Eotvos Lorand University, Pollack tér 10., Budapest-1088, Hungary | <h56kab@ella.hu>

ABSTRACT

This paper revisits a more than twenty-five-year old idea of G. F. Jenks and W. R. Tobler about the relationship between accuracy, information and map complexity of choropleth maps. The problem of regionalization (*sensu* aggregation) is treated within a spatial statistical context. For a map with N regions to be aggregated into G groups, nonspatial hierarchical classification schemes disregard spatial pattern and are prone to lead to non-contiguous classes (i.e., each group may consist of a large number of patches). Restricting merges during clustering according to neighborhood-topological relationships rewards contiguous patches of classes, but may impose too strict, potentially misleading, constraints. To obtain more efficient, less complex aggregate representations (e.g., maps) we propose to evaluate efficiency by a modified version of Akaike's information criterion: AIC' = (-2 x loglikelihood + 2 x number of patches). It follows from the general principle of model selection, by minimizing the sum of fitting error and some measure of model complexity, Socio-economic, environmental and simulated data are used to highlight the characteristics of this approach, which appears particularly useful when no additional information is available to select the number of groups.

INTRODUCTION

The art and science of creating beautiful and meaningful maps based on some two-dimensional distributions has attracted people for several hundred years. In particular, major efforts have been focused on creating "the" spatial/cartographic analogy of classification; i.e., to put the N elements (data representation units, DRUs) of a two-dimensional lattice into G<<N "spatial groups". Such tasks often emerge in studies of socio-economic variables (e.g., defining wealthy/poor neighborhoods), in environmental studies (e.g., finding locations of suitable habitats) and in many other geographically-oriented fields.

Considering the frequent occurrence and diversity of applications of such tasks, it is not surprising that several detailed studies and overviews have focused on the series of "map-making" decisions and their optimization. Classical *cartographic* treatises, typically under the "error and classification of choropleth maps" keywords, can be found in Jenks and Caspall (1971), Monmonier (1973), Stegena and Csillag

(1986) and, in textbook format, in Robinson et al. (1995, p.517). More analytical approaches to similar problems are dealt with in *spatial statistics* (with widespread applications in econometrics, epidemiology, soil science) generally under the "aggregation and the modifiable areal unit problem" headings, for example, in Unwin (1981), Haining (1990) and Cressie (1993). A somewhat closely related array of techniques have emerged in *image processing* usually referred to as "image segmentation" (see, e.g., Schowengerdt, 1983, Kertesz et al., 1996). Several reports recognized the relationships, and interactions, among these procedures and some attempted to define a more general framework for "spatial data representation" (e.g., Maguire et al., 1991). Within the context of geographical information systems (GIS), often linked with statistical software packages, "spatial grouping" is also a frequently occurring common task, even if it is performed with diverse goals ranging from illustration, detection and verification of spatial patterns, optimization of visual and/or functional representation.

The real impetus for this paper, however, is an intriguing idea illustrated on the last, an apparently neglected, figure (see Figure 1) from the seminal paper of Jenks and Caspall (1971).

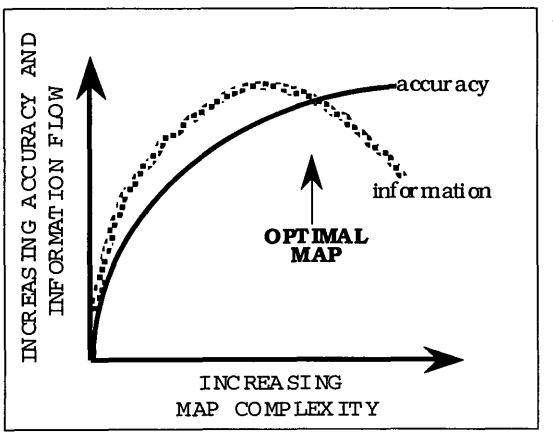


FIGURE 1.
Relationship between accuracy and information flow on maps; redrawn from to Jenks and Caspall (1971). The accuracy vs. complexity relationship was supported by empirical data; the information vs. complexity relationship was based on Waldo Tobler's personal communication. Interestingly, the choice of an "optimal map" does not correspond to either maximum information flow, or to maximum accuracy.

This figure seems to suggest more challenges than conclusions:

- How does one measure information flow as a function of map complexity?
- What evidence supports the shape of the "information vs. complexity" curve? What determines its position along the complexity axis?
- What evidence supports the existence of a unique intersection of the "information vs. complexity" and the "accuracy vs. complexity" curves? What does its position depend on?
- What algorithm is suitable for finding the optimal map?
- Assuming there is a unique intersection, what justifies the selection of the marked "optimal map" *instead* of maximum information (or maximum accuracy)?

(RE)DEFINING THE PROBLEM: MEASURING INFORMATION LOSS

Let us define the problem as follows. The object set (T) consists of N units (e.g., polygons in a coverage, pixels in an image). The data set (Y) is a K-dimensional variable observed at each element of T. $P^{(G)}$, a partition of T, consists of G collectively exhaustive disjoint classes. Each class covers a region that may consist of one or more patches. The special regionalization where all classes are spatially contiguous (i.e., the number of classes (G) equals the number of patches (R)) is called segmentation. Note that the finest partition, the only N-partition is $P^{(N)}$, and the coarsest partition, the only 1-partition is $P^{(1)}$, and the definition of fine/coarse (and finer/coarser) is the usual. Let D(P) denote the *discrepancy* between a selected regionalization and the observed phenomenon. In light of general statistical model selection (Linhart and Zucchini, 1986), our model of choice should be parsimonious, i.e., it should not have more parameters than the ones which can be reliably estimated, for example, to avoid "overfitting". Thus, discrepancy consists of two parts: one due to *approximation*, and another due to *estimation*. The first component, $D_A(P)$, practically measures model complexity, and is often completely neglected. The second component, $D_E(P)$, measures the goodness of fit between the sample and the chosen (approximating) model. In the above outlined classification example it is the "loss of information due to grouping", and it is most commonly measured by the expectation of the negative loglikelihood. This leads us to Akaike's information criterion as a measure of discrepancy (Akaike, 1973):

AIC = (2 x number of parameters -2 x loglikelihood).

Our task is to minimize the discrepancy over the set of all possible partitions P*. Note that P* consists of potentially very large number of elements (2^N), thus there is no real chance to find the exact solution. Clustering procedures, therefore, are typically confined to some subset of P* while minimizing D(P). An acceptable way to avoid the problem of comparing, and thus choosing from, models of different complexity by computing $D(P)=D_A(P)+D_E(P)$, is to set G, i.e., to reduce the problem to finding $P^{(G)}$, the G-class map, with the smallest $D_E(P)$. Cromley (1996) provides an extensive recent review of comparing different "estimation discrepancies" with given number of classes.

In this paper we will consider the problem when the number of classes (G) is not known. Hierarchical clustering algorithms, for example, are suitable to scan a subset of P*, the monotone aggregating (coarsening) sequence of $P^{(N)}$, $P^{(N-1)}$, ..., $P^{(1)}$, i.e., they start from the finest partition (all elements form a separate class) and the number of classes decreases by one in each step (by merging two classes) until the coarsest partition, $P^{(1)}$, is reached. The algorithms differ from each other in the way they decide which two classes to merge. The most common choice for $D_E(P)$ is the ratio of within-groups variance/total variance. The value of $D_E(P)$ can be regarded as a measure of separation of partition P. The Ward-method of clustering (Ward, 1963) in each step selects the pair of clusters to merge by minimizing the increase in the

above defined "estimation discrepancy" leading to the monotone increasing sequence of $D_E(P)^{(N)}$, ..., $D_E(P)^{(1)}$. Note that there is no guarantee that any member of this sequence is close to the minimum of $D_E(P^*)$.The likelihood in the case of a simple product MVN(μ,σ) is:

$$\mathcal{L}(\mu,\sigma,Y) = \text{const} \times \exp\{-(1/2) \times (1/\sigma^2) \times \Sigma_n[y_n-\mu_n]^2]\}$$

for which the (-2 x loglikelihood) reduces to the within-groups sum-of-squares if σ^2=1. As Jenks and Caspall (1971) also note, accounting for $D_E(P)$ only during aggregation, one would always choose the map with each DRU being a separate class, because $D_E(P^{(N)})$ provides the "best" separation by *value*. Following from AIC, the discrepancy due to approximation, $D_A(P)$, should equal the number of classes (G).

In geographical applications, when judging whether a partition is "good" or not, one is frequently concerned with pattern, the separation by *location* as well. Assuming that we are looking at "dirty pictures", i.e., realizations, where some "crisp" regions are blurred by noise, it is essential to use methods which are robust in "finding" the regions, thus accounting for $D_A(P)$ as well. One approach in this direction is the restriction of the subset from which a clustering algorithm chooses classes to merge according to neighborhood-topological information. Such "patch"-versions can be implemented for any hierarchical clustering, similarly to several region-growing algorithms developed in image analysis (Landgrebe, 1980). If we restrict merges to neighbors, the number of classes (G) equals the number of patches (R) in each step.

AN EXAMPLE

Let us illustrate how these measures of discrepancy work with a simple example. Figure 2a. shows a simple map of 64 DRUs with three classes (0, 8 and 9 represent values), which form three "crisp" regions, or patches. Figure 2b. and 2c. are "standard" cartographic representations with three equal-count and three equal -interval choropleth maps, respectively. Aggregating this map with Ward-clustering and its "patch"-version, we can plot the within-group sum-of-squares (SSQw), the number of classes and the number of patches for each iteration (Figure 3.).

To generate a measure of information (see Figure 1.), both clustering procedures can be characterized by AIC (in this case SSQw+2 x number-of-classes); cAIC denotes the case of Ward clustering and pAIC denotes the case of Ward_patch clustering (Figure 4.). Note that cAIC practically serves as a stopping rule, but it "stops" a little bit "early". Therefore, we propose to investigate a modified version, cAIC'= SSQw+2 x number-of-patches for the Ward-clustering, because it retains essential information about the pattern, while it is not prone to the restrictions of Ward_patch. The minimum of the AIC-plots corresponds to the "minimum information loss" due to the model, and such measures are particularly useful in comparing the nested series of models generated by hierarchical clustering.

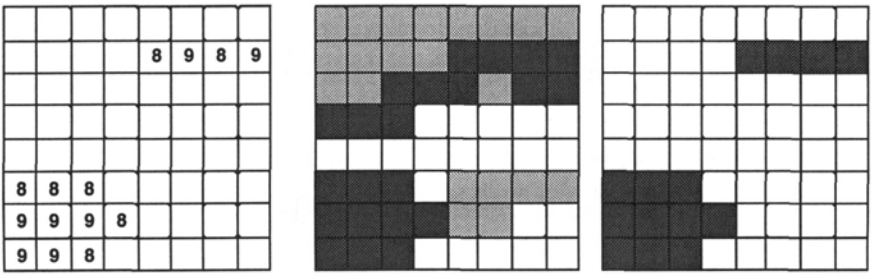


FIGURE 2.

Sample map (a) with three classes forming three "crisp" regions (0, 8 and 9 are values). Its three-class equal-count (b) choropleth map represents seven patches and its three-class equal-step (c) choropleth map represents three patches.

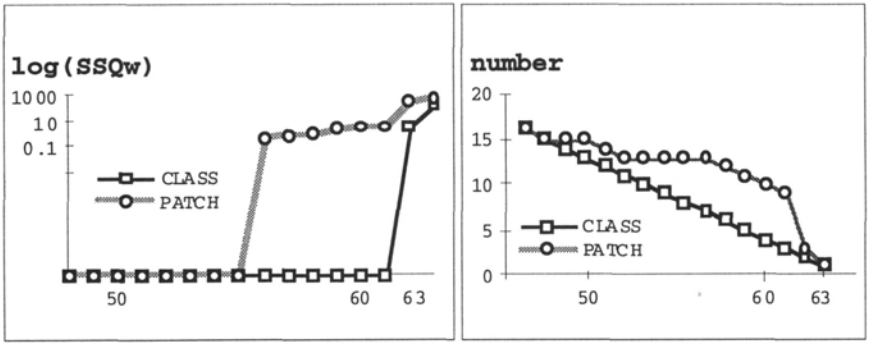


FIGURE 3.

SSQw (left) and the number-of-classes/patches (right) for the last 16 iterations of Ward and Ward_patch clustering of Figure 2a. SSQw is the discrepancy due to estimation, and the number-of-classes or the number-of-patches is the discrepancy due to approximation.

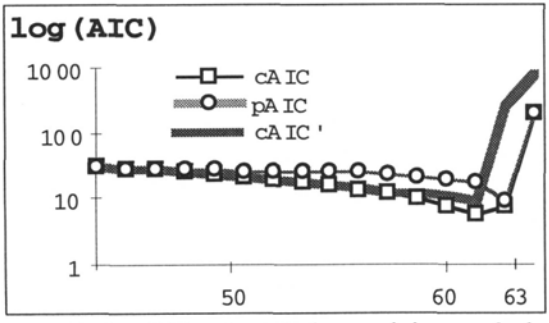


FIGURE 4.
Plot of measures of "information loss" for the last 16 iterations of Ward (cAIC, cAIC') and Ward_patch (pAIC). The minimum of these functions corresponds to minimum discrepancy between the model and the observation.

Both cAIC and pAIC have minima at 3-classes/9-patches (i.e., each value in Figure 2a. forms a separate class), while the minimum of cAIC' coincides with 2-classes/3-patches. In other words, both cAIC and pAIC seem "to overfit" resulting in a tendency to reduce SSQw at a cost of greater number of patches. One can apply

cAIC' as an alternative stopping rule for agglomerative hierarchical clustering of geographical phenomena.

A SIMULATION EXPERIMENT

To investigate the behavior of these measures of discrepancy we have conducted a simulation experiment. Because we are interested in deriving some information about spatial pattern (c.f. regionalization), noise with various levels of spatial autocorrelation was added to Figure 2a, and these realizations were aggregated using both clustering algorithms (Ward and Ward_patch). The spatial structure of noise was controlled by the conditional autoregressive (CAR) parameter ρ (Cressie, 1993):

$$\mathbf{Y} \approx \mathrm{MVN}[\mu, \sigma^2(\mathbf{I}-\rho\mathbf{W})^{-1}]$$

where $w_{ij}=1$ for neighbors (0 otherwise), and we set $\mu_{ij}=0$ and $\sigma^2=1$. Fifty realizations were analyzed for ρ set to 0.0, 0.1, 0.2, and 0.245, respectively. Table 1. summarizes the results for the two extreme values of ρ, and Figure 5. shows example outputs.

TABLE 1.
Summary of fifty simulations for extreme values of spatial autocorrelation. Rows contain mean values and standard deviations for various measures of aggregation quality. Columns refer to different merging and stopping rules for Ward clustering.

ρ=0	cAIC		pAIC		cAIC'	
minimum	21.37	1.58	48.45	5.45	54.09	6.87
class	7.57	0.79	14.43	3.87	5.00	1.53
patch	35.14	5.05	47.86	6.52	15.29	6.50
iteration	56.43	0.79	49.43	3.69	59.00	1.53
SSQw	6.22	1.82	19.31	4.73	23.52	7.90

ρ=0.245	cAIC		pAICp		cAIC'	
minimum	22.90	2.03	46.26	4.79	55.04	4.92
class	8.14	1.35	15.14	3.44	5.00	1.29
patch	33.29	5.47	46.86	5.90	16.29	3.35
iteration	55.86	1.35	48.86	3.44	59.00	1.29
SSQw	7.72	3.20	18.64	8.47	26.21	3.80

One would expect that as the spatial autocorrelation of noise increases the higher the chance to mislead the clustering algorihtm by forming "artificial" patches.

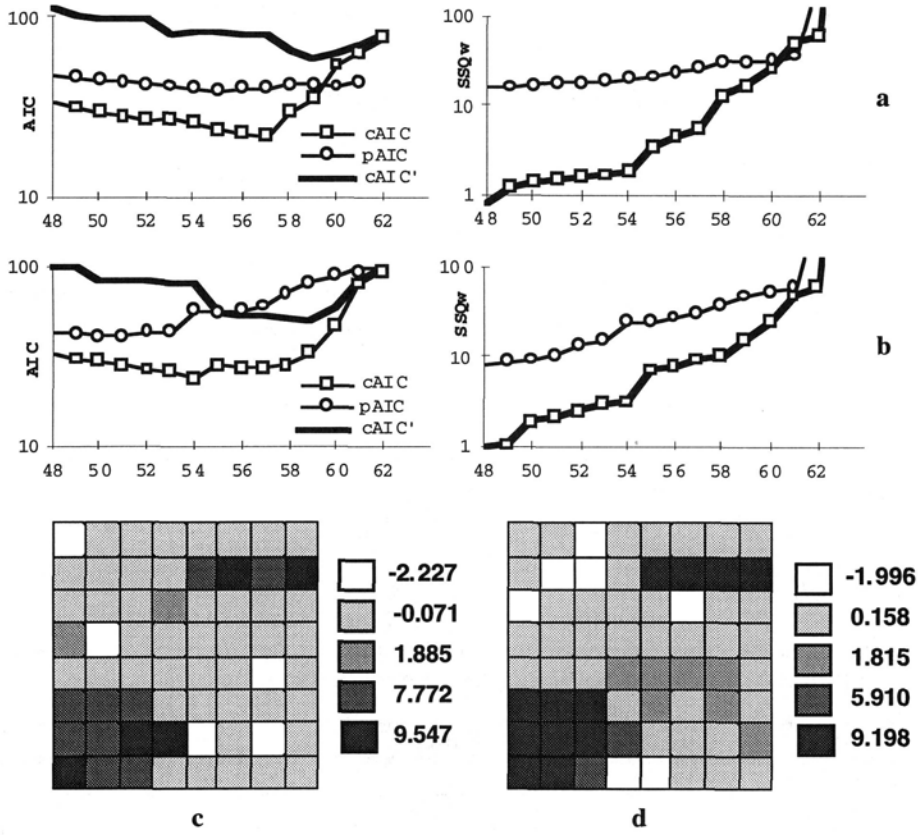


FIGURE 5.

Sample outputs from the simulation-aggregation study. Values of SSQw and AIC for a "typical" run for the sum of Figure 2a. and spatially not correlated noise (a) and spatially highly correlated noise (b). The corresponding 5-class maps corresponding to the minimum of cAIC' are shown on (c) and (d), respectively.

Clearly, the cAIC' stopping rule is the most resistant to the increasing ρ; its average choice for the number of classes remains the same (5.0), while both other measures tend to choose more classes and even more patches ($D_A(P)$), at each ρ. Of course, it comes at a cost of greater $D_E(P)$; the values of SSQw are significantly higher than for the other two measures. It is important to note that in real applications, typically, there is no information about the relationship between the amount of noise and the nature of boundaries, therefore, there is always a chance to overfit to "islands" (when using cAIC), or to "awkward patches" (when using pAIC).

TOWARD APPLICATIONS

The implementation of using any of the above described measures in geographical analysis is relatively simple in commercially available GIS software.

Below, two very different mapping problems are used to illustrate the applicability of the findings where regionalization is the task. We have intentionally selected examples, where no *a priori* information can be easily used.

Case-1: Regions of high acid deposition in the northeast US are intensively studied to understand and predict its impact on the soil-water-plant systems. Since long-term acid deposition measurements are only sporadically available, elevation has been used as a surrogate for the amount of wet acid input into lake ecosystems (for example, for defining sampling strata).

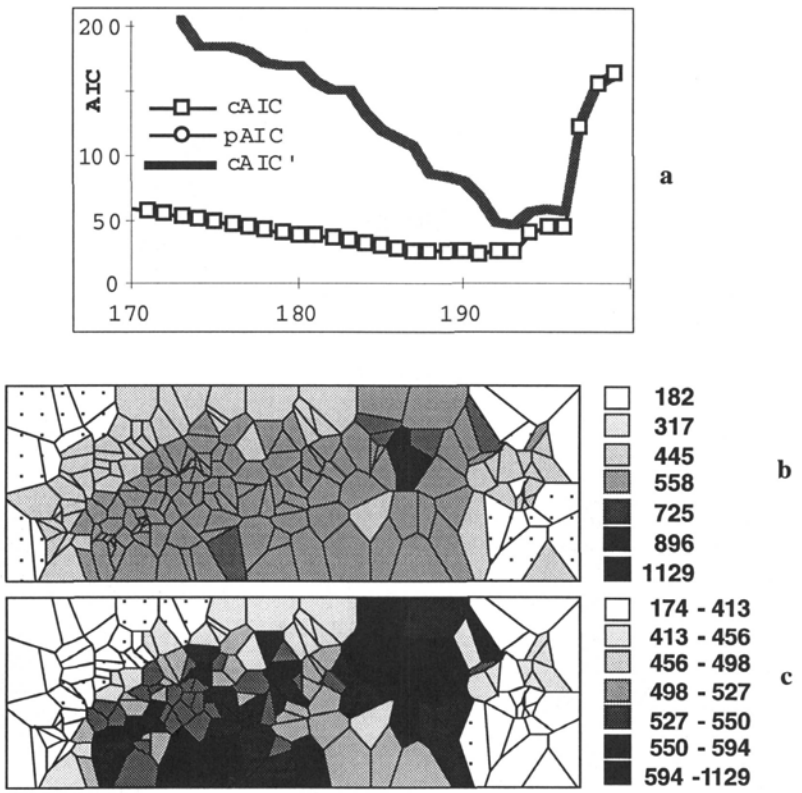


FIGURE 6.

An east-west cross section of the Adirondack Mountains, NY, with the Voronoi-polygons for 200 lakes from the Adirondack Lake Survey. A relatively smooth variable, elevation (m) is recorded as a surrogate for acid deposition to delineate variously impacted areas. Discrepancy values (last 30 iterations shown, (a)) for the different clustering procedures coincide at 7 classes (b). The 7-class equal-count choropleth map (c) gives a vastly exaggerated impression.

A subset of 200 lakes from the Adirondack Lake survey along the major elevation gradient is used in this test. Figure 6. summarizes the results, which indicate that

even in case of relatively smooth variation, traditional choropleth mapping (simple histogram-partitioning) can lead to quite misleading results.

Case-2: In urban socio-economic research, the delineating regions (e.g., for market, services, voting behavior) often aims to identify "areas of action" or "areas of influence". Below, we show an example using percentage of unemployment based on 121 enumeration areas in the Greater Toronto Area. The three clustering procedures result in significantly different regionalizations (Figure 7.). Because of the small, intensively segmented southwestern section, according to pAIC, cAIC and cAIC' one would select 13 classes (77 patches), 8 classes (72 patches) and 3 classes (19 patches), respectively. The "closest" equal-step choropleth map is shown for comparison.

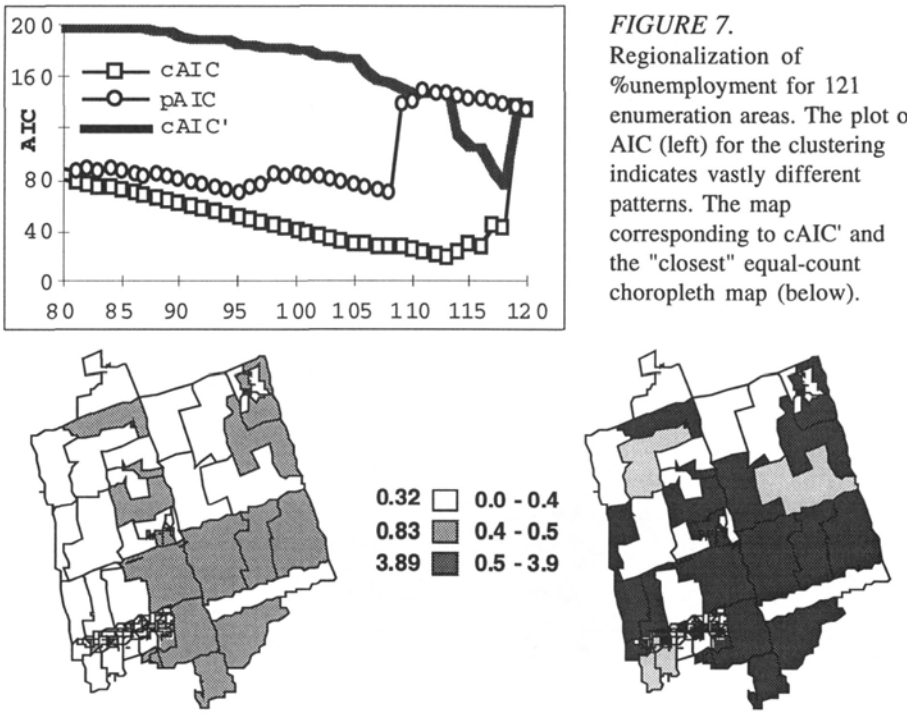


FIGURE 7.
Regionalization of %unemployment for 121 enumeration areas. The plot of AIC (left) for the clustering indicates vastly different patterns. The map corresponding to cAIC' and the "closest" equal-count choropleth map (below).

CONCLUDING REMARKS

There are many conceptual models of geographical regions. When landscapes are represented by some variables attached to some data representation units, spatial statistical tools can be applied to "finding" homogeneous regions, especially when no other ancillary information (constraint, requirement) is available. Within this context we revisited the proposition of optimizing "information flow" (Jenks and Caspall, 1971) and compared three different measures of it using hierarchical (Ward) clustering.

In a simulation study, accounting for the spatial autocorrelation of noise, we found the modified, topologically sensitive Akaike information criterion a robust measure to avoid "overfitting" and moderately "reward" contiguous patches. The immediate next step should be to implement the CAR-based likelihood in AIC. The proposed type os measure is relatively simple to implement in commercially available software, at least to be used as guidelines in creating choropleth maps. It is also an advantage, that the computation is straightforward to extend to the multivariate case (i.e., regionalization based on more than one variable).

ACKNOWLEDGMENT

We thank N. Fogarasi, an undergraduate at the University of Toronto, for programming. Parts of this research was conducted while S. Kabos visited Toronto for which the financial support of NSERC is gratefully acknowledged.

REFERENCES

Akaike, H., 1973., Information theory and an extension of maximum likelihood principle. Second Int. Symp. Information Theory (eds. B. N. Petrov and F. Csaki), pp. 267-281. Akademiai Kiado, Budapest.

Cressie, N. A., 1993. Statistics for spatial data., J. Wiley, New York.

Cromley, R. G., 1996., A comparison of optimal classification strategies for choroplethic displays of spatially aggregated data. Int.J. Geog. Info. Sys. 10:405-424.

Haining, R. P., 1990. Spatial data analysis in the social and environmental sciences. Cambridge University Press, Cambridge.

Jenks, G. F. and Caspall, F. C., 1971., Error on choroplethic maps: definition, measurement reduction. Annals of the Assoc. Amer. Geog. 61:217-244.

Kertesz, M., Csillag, F. and Kummert, A., 1996. Optimal tiling of heterogeneous images. Int. J. Remote Sensing 10

Linhart, H. and Zucchini, W., 1986., Model Selection. J. Wiley, New York.

Maguire, D., Goodchild, M. and Rhind, D., 1991. Geographical information systems: Principles and applications. J. Wiley, New York.

Monmonier, M., 1973., Analogs between class-interval selection and location-allocation models. The Canadian Cartographer. 10:123-131.

Robinson, A.H., Morrison, J. L., Muehrcke, P.C., Kimerling, A.J. and Guptill, S.C., 1995. Elements of cartography. J. Wiley, New York.

Schowengerdt, R. A., 1983. Techniques for image processing and classification in remote sensing. Academic Press, New York.

Stegena, L. and Csillag, F., 1986., Statistical determination of class intervals for maps. The Cartographic Journal 24:142-146.

Unwin, D., 1981., Introductory spatial analysis. Methuen, London.

Ward, J. H., 1963., Hierarchical groupings to optimize an objective function. J. Amer. Stat. Assoc. 58:236-244.

REASONING-BASED STRATEGIES FOR PROCESSING COMPLEX SPATIAL QUERIES

Ilya Zaslavsky, Assistant Professor,
Department of Geography, Western Michigan University, U.S.A.

ABSTRACT

This paper describes potential strategies for analyzing complex spatial queries in multi-layer vector GIS. The purposes of such analysis are (1) to reduce the size of the query, still providing acceptable accuracy, and (2) to provide information to the user about how the query should be reformulated to obtain an acceptable result. Several reasoning-based strategies for the reduction of query size are considered: finding reasoning chains which lead to the most accurate available approximation of a query; filtering out least significant categories, identifying the most sensitive elements in a query which could produce best gains in accuracy once re-specified. Since elements in a complex query, including categories, relations between categories, and spatial context, can be specified to a given certainty, the problem involves reasoning with imprecise premises, and certainty propagation. The task is formalized within the framework of determinacy analysis and logic which provide a computational solution for the accuracy of a corollary statement (query result, in our case) based on such "imperfect" premises. A series of experiments demonstrate the dependence of the query accuracy on the absolute values and on the degree of certainty in definitions of each category and relation in the query.

INTRODUCTION

Processing complex spatial queries is one of fundamental capabilities of Geographic Information Systems (GIS). Formulation of query languages encompassing a wide variety of spatial analytical tasks has been a subject of extensive recearch in recent years (Ooi, 1990; Langran, 1991; Tomlin, 1990; Egenhofer, 1992, etc.) Responding to a query can be fairly straightforward, when it involves only an attribute database search. However, common queries in cartographic modeling may involve more than one attribute, and require overlay of several map layers, or some other geometric processing. Consider, for example, a query "select areas in parks within the city, such that there is a lake within the park, and also the area has slopes not greater than 5% and soils of a given type". A direct way to resolve such a query is to overlay maps of parks, lakes, city boundaries, slopes, and soils. Though each subsequent overlay

deals with smaller area, the solution may take a lot of computer resources. Besides, a rigid following the definitions of the categories and relations may result in a zero answer, without providing any information about how the query should be reformulated, and thus making the "what-if" scenario of geographic analysis with GIS a long and frustrating experience. It is important, therefore, to resolve such a query, or some aspect of it, in a way that (1) minimizes the processing time required to report the results, and (2) suggests how to improve the query by re-specifying its elements.

The fact that each of the elementary query components can contain uncertainty, requires their formal modeling as uncertain statements, and modeling error propagation in combinations of such statements. This paper investigates how such complex queries can be decomposed and optimized, using a set of analytical and reasoning techniques known as determinacy analysis and determinacy logic (Chesnokov, 1990; Zaslavsky, 1995). Determinacy logic allows to estimate the binary truth values for syllogisms with uncertain premises, and, conversely, to propagate certainty bounds in reasoning chains. We will consider reasoning-based estimates of the area covered by a combination of categories to be reported by a complex query. The paper starts with a formalization of uncertainty propagation in a complex query, as a reasoning problem. Then, we compare different methodologies for reasoning about elements in such query. Finally, a series of experiments are described showing the strategies for query improvement.

UNCERTAINTY IN A COMPLEX QUERY, AND ITS FORMALIZATION

The complex query described above, has several important properties. The results reported by a query depend on both definitions of categories (park, lake, city, soils), and relationships ("within" and "intersect", in this case, see Egenhofer and Franzoza, 1991, and subsequent works on qualitative spatial reasoning on description of other topologically distinct spatial relations). Uncertainty inherent in such definitions may be greater than uncertainty associated with formal processing of geographic data in GIS, and it should be taken into account during the translation of common-sense geographic circumstances into a formal language of GIS queries.

It may be possible to resolve a complex query with acceptable accuracy (within user-defined certainty thresholds) without performing an overlay. If a sufficient amount of information about previous queries has been accumulated in the system, new queries can be resolved with the help of a reasoning engine.

Let's consider a query based on elementary categories "a" and "c" from layers A and C, respectively. Each of the categories is specified with certain accuracy, that is, areal proportions of "a" and "c" in the entire area, *P(a)* and *P(c)*, are such that $\omega_1 \le P(a) \le \theta_1$, and $\omega_3 \le P(c) \le \theta_3$, where ω and θ are some numbers in the [0,1] interval (here and below I follow the notation of Chesnokov, 1990). The task is to respond to a query about the area in overlay of "a" and "c".

Beyond an obvious (and seldom useful) solution

$$\max\left\{\begin{matrix}0\\P(a)+P(c)-1\end{matrix}\right\} \le P(ac) \le 1, \quad \text{or}$$
$$\max\left\{\begin{matrix}0\\\omega_1+\omega_3-1\end{matrix}\right\} \le P(ac) \le 1 \qquad (1)$$

the task can be described as a quantitative reasoning problem, in which auxiliary information is used to better specify the relationship between *"a"* and *"c"*. Suppose we don't know the relation between *"a"* and *"c"*, but we have accumulated information about the relations between these two categories, and categories from other layers in the same database. Let's call such other category *"b"* from layer B, and characterize its uncertainty as $\omega_2 \le P(b) \le \theta_2$, similarly to the specification of categories *"a"* and *"c"* above. Each of the relations, *(a→b)* and *(b→c)*, may be also uncertain, i.e. the areal proportions of combinations of *"a"* and *"b"*, *"c"* and *"b"*, respectively, are described as:

$$r_{12} \le P(ab)/P(a) \le s_{12} \quad \text{and} \quad r_{23} \le P(bc)/P(b) \le s_{23}$$
$$r_{21} \le P(ab)/P(b) \le s_{21} \quad \text{and} \quad r_{32} \le P(bc)/P(c) \le s_{32} \qquad (2)$$

The task then is to find such intermediate category "b" so that the syllogism

$$(a \to b) \text{ and } (b \to c) \Rightarrow (a \to c) \qquad (3)$$

is true, and relation *(a→c)* is accurate within the preset limits

$$r_{13} \le P(ac)/P(a) \le s_{13}$$
$$r_{31} \le P(ac)/P(c) \le s_{31} \qquad (4)$$

By obtaining a narrow estimate of *P(ac)/P(a)* and *P(ac)/P(c)*, we would approximate a query involving overlay of *"a"* and *"c"*.

ALTERNATIVES FOR UNCERTAINTY PROPAGATION IN COMPLEX QUERIES

The desirable properties of a spatial reasoning engine for the problem described above are: (1) ability to process inexact premises (which makes the machinery of Boolean algebra inapplicable); (2) topological "conformance", or description of uncertainty as deviations from topologically distinct cases requested by most kinds of queries; (3) ability to interpret the reasoning outcome as proportions of areas rather than abstract certainty values, and (4) ability to handle different kinds of relationships between premises, including transitivity and multiple evidence. Below, we briefly characterize some available reasoning schemes from the perspective of these desired properties.

Probabilistic reasoning

The most common way to solve the problem described above is to interpret the proportions of areas as probabilities, and apply some probability propagation technique (like Bayesian combination of beliefs). Some of the problems associated with this approach are: (1) large size of a completely specified model where knowledge of each category is conditioned on knowledge of all other categories, and all of their combinations. This size is typically lowered by using the conditional independence assumption, which is often not true for geographic data; (2) transitivity as a fundamental element of material-implication interpretation, is shown to be wrong in AI systems based on Bayesian propagation (Pearl, 1988), and (3) arbitrary assignment of prior probabilities. The critical question is whether the very interpretation of empirical relative frequencies and areal proportions as probabilities is justified. Following Kolmogorov (1951), for example, we can consider probabilities as both purely mathematical objects (first section of his famous "Foundations of the Theory of Probability", 1933), and empirical frequencies in von Mises's interpretation (second section of the same book). From this perspective, empirical objects should be treated as probabilities if they conform with the axiomatics of probability calculus. Practically, in order to make the transition to probability, it is necessary to specify a random process, and a homogeneous probability field. None of these requirements are typically satisfied for common data layers in GIS.

Fuzzy reasoning

Fuzzy representation of map categories is useful for modeling boundary uncertainty (Burrough, 1989; Heuvelink and Burrough, 1993), and for processing multiple statements with uncertainty. However, fuzzy membership is different from certainty of statements which describe relations between categories as proportions of areas. Lack of empirical basis of membership

grades, and axiomatic propagation of membership values, make fuzzy reasoning inadequate in the tasks of empirical analysis of complex queries to traditional map information. Converting proportions of areas into fuzzy membership grades would be another interpretational leap which is difficult to justify.

Determinacy logic

This approach, developed by Chesnokov (1984, 1990), focuses on processing empirical conditional frequencies without interpreting them as either probabilities or fuzzy membership grades. On the elementary level, Determinacy Analysis focuses on statements in the form "*IF a THEN b*" called determinacy statements, or $(a \rightarrow b)$, and accompanied by values of statement *accuracy* (proportion of *"b"* in *"a"*, or $I(a \rightarrow b) = P(BA)/P(A)$), *completeness* (proportion of *"a"* in *"b"*, or $C(a \rightarrow b) = P(BA/P(A))$, and *context* (portion of the database for which the statement is examined). The main formal object of Determinacy Logic is *determinacy syllogism*, a statement connecting two determinacies, $(a \rightarrow b)$ and $(b \rightarrow c)$, to produce corollary $(a \rightarrow c)$. Its general analytic solution, for arbitrary lower and upper bounds on the definitions of categories and relations, has been obtained by Chesnokov (1990). The advantages of determinacy reasoning over other reasoning systems when applied to data in GIS, include: (1) material-implication interpretation of certainty measures (i.e., the resulting measures of uncertainty can be expressed in proportions of areas rather than in abstract units); (2) a computational solution for bounds propagation is provided, versus axiomatic approaches of other logical systems; (3) the conditional independence assumption of Bayesian beliefs propagation is not employed; (4) transitivity syllogisms are allowed, by contrast to AI systems based on Bayesian schemes; (5) qualitative reasoning about topological spatial relations can be considered as its general case.

Within the determinacy approach, responding to complex queries can proceed as follows (figure 1). Once the user specifies a spatial query about relationship $(a \rightarrow c)$ in context k, the system searches a previously accumulated meta-database of relationships between *"a"*, *"c"*, and categories from other layers, for such intermediate category *"b"*, that combination of $(a \rightarrow b)$ and $(b \rightarrow c)$ produces the most accurate and narrow estimate of $(a \rightarrow c)$. If the estimated accuracy of the query is not acceptable, the actual polygon overlay has to be performed, with a direct computation of query characteristics. The results of this overlay are appended to the database of relationships, to be used in estimating future queries.

Each record in the database of relationships between layers represents a description of a determinacy statement; its structure can be as follows: (1) context of determinacy k (locational, incidence, neighborhood, directional);

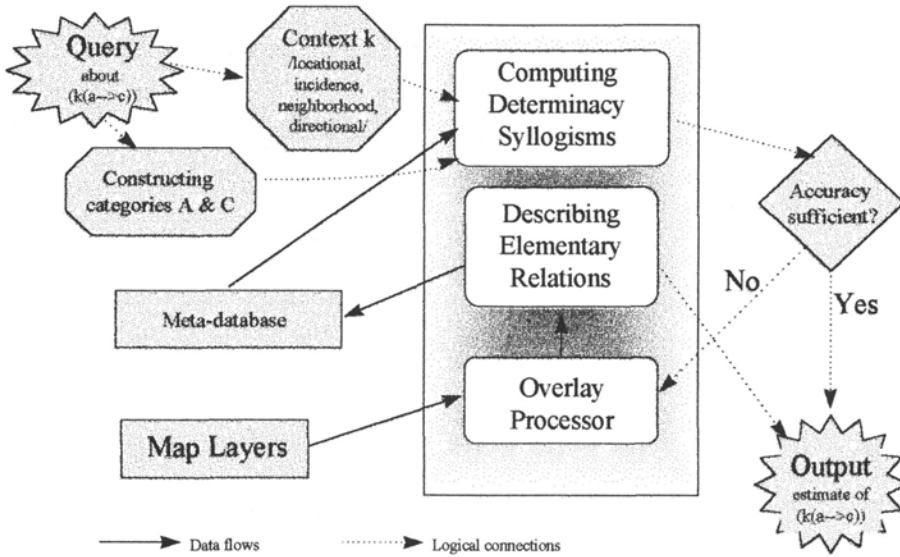


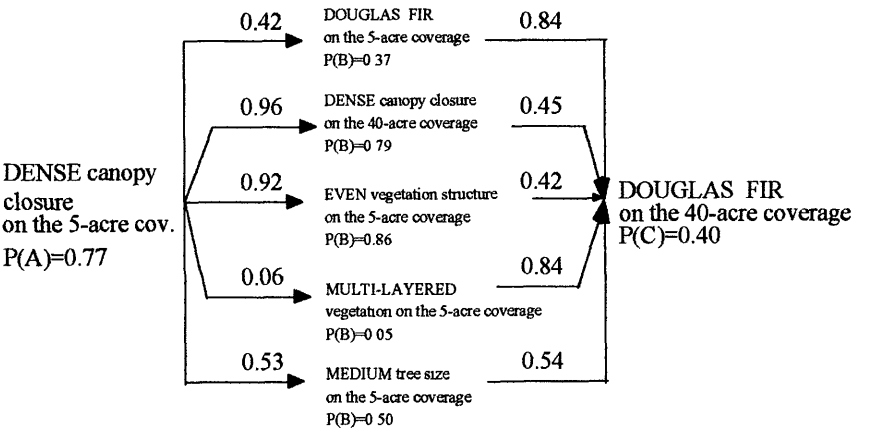
Fig. 1. Query implementation in a system based on determinacy logic

(2) the "argument" of determinacy *(a→b)*, *"a"* (a single category, or a combination of categories); (3) the "function" of determinacy *(a→b)*, *"b"* (a single category, or a combination of categories); (4) *P(a)* - proportion of the study area covered by category, or combination of categories, *"a"*, in the context *k* defined in the first field; (5) *P(b)* - - proportion of the study area covered by category, or combination of categories, *"b"*, in the same context; (6) accuracy of determinacy *(a→b) = Area (a & b)/Area(a)*; (7) completeness of determinacy *(a→b) = Area (a & b)/Area(b)*. The information in this table can accumulate in the self-learning process during regular work with the dataset. Besides, the dataset can be left in a "training" regime, when the program builds a meta-database for given contexts, or for certain layers. Eventually, sufficient information accumulates and starts to produce reasonable accuracies of corollary statements.

Currently, this approach is implemented as a loosely coupled set of programs. Arc/Info is used to formulate and process queries, then the database is dumped into a text file and processed with the LOGIC module of the determinacy analysis package. This module is used in examples and computations below.

Figure 2 shows a computational example of this scheme with the data from the Klamath Province Vegetation Mapping Pilot Project (Final Report..., 1994). The chain producing the most narrow response to a query about the

combination of *"a"* ("dense canopy closure on the coverage with 5 acre minimum resolution") and *"c"*("Douglas Fir on the 40-acre coverage"), includes "dense canopy closure on the 40-acre coverage" as the intermediate category *"b"*. This reasoning produces the area estimate in overlay between "a" and "c" as between $2.881 * 10^6$ and $3.566 * 10^6$ acres (the actual area is $3.235 * 10^6$ acres), i.e. the accuracy is within 10%.



Relation	Lower bound	Upper bound	Range
1	0.343	0.516	0.173
2	0.403	0.499	0.096
3	0.272	0.516	0.244
4	0.219	0.516	0.297
5	0.234	0.516	0.282

Fig. 2. An example of reasoning-based estimate of query results (source of data: Klamath Province Vegetation Mapping Pilot Project, 1994). The value on each arrow is accuracy of corresponding determinacy.

REFORMULATION OF A QUERY

Now suppose that the accuracy of the estimate obtained above is below the user's expectations, i.e. the area under the combination of categories "a" and "c" is not in the interval specified by inequalities (4). The task then is to inform the user about those elements of the query that need reformulation in order to approach the desired accuracy in an optimal fashion.

For the simplest case, the graphic idea of a solution is shown in Figure 3. In this case, where $\omega_i = \omega$; $\theta_i = 1$; $r_{12} = r_{23} = r_{13} = \mu$; $s_{12} = s_{23} = s_{13} = 1$, the lower bound on accuracy ν of the corollary statement (Chesnokov, 1984) is:

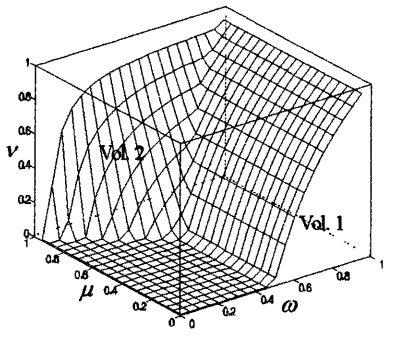


Fig. 3. The solution space for the simplest case is composed of two volumes, depending on the context (first), and on both context and the premises (second).

$$\nu = \max\begin{Bmatrix} 0 \\ 2 - 1/\omega \\ 1 - (1-\mu)(\mu + 1/\omega) \end{Bmatrix} \quad (5)$$

The solution space is formed by two volumes, the first one depending on ω only, and the second depending on both ω and μ. For any point specified in coordinates (ω, μ, ν) beyond these two volumes, it is possible to determine its distance to each of the volumes. It is assumed that following the shortest distance to the area where the syllogism is true, translates into suggested changes to parameters of the query. For example, if the point in question is closer to the first volume, it makes sense to redefine the categories involved in the query (either the context of the category, or its width, or both), and vice versa.

Below, we show the results of numeric experiments with the general solution of determinacy syllogism, for arbitrary ω_i, θ_i, r_{ij}, and s_{ij}. The purpose of the experiments is to demonstrate which parameters (absolute values of the context and the accuracy of the premises, and their certainty intervals) need priority improvement to make the syllogism correct. The results are shown in Figure 4. The contour plot on the left panel shows the dependency of the lower accuracy bound of the corollary statement upon the context ω (horizontal scale), and upon the accuracy of the premises μ (vertical scale) in a query, with 1%-wide uncertainty of the context. For the most part, the increase in the context values does not lead to any gain in accuracy until ω reaches 0.5 for premises with accuracy 0.5 and higher. The accuracy of the query rapidly increases when the values approach $\omega = 0.5$ while the accuracy of the premises remains low. In this case, which corresponds to situations close to maximum avoidance between categories *"a"* and *"c"*, the emphasis on narrowing the context would lead to dramatic increase in accuracy of the query. If the values of the context are fairly low (0.1 - 0.5), and accuracy of the found premises is above 0.6, only further increase in the accuracy of premises would pay off with higher accuracy of the composite query.

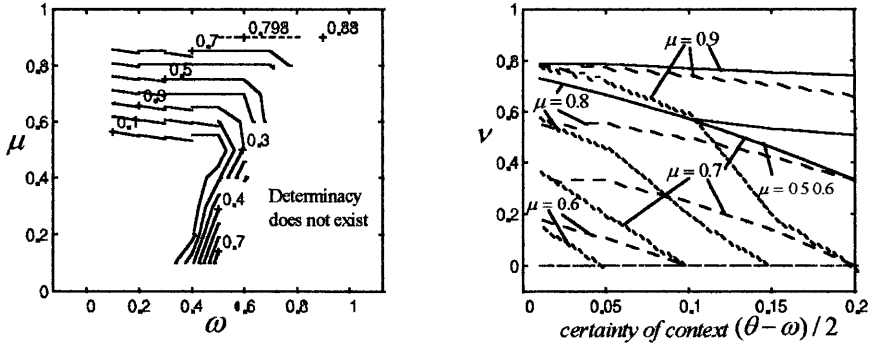


Fig. 4. Dependency of query accuracy on its components: on the context of the categories (specified to 1% certainty) and accuracy of the premises (left panel); on the width of certainty interval for different absolute values of the context and accuracy of premises (right panel).

The second plot demonstrates the dependence of the query accuracy on the width of uncertainty interval for the context. With the decrease in the uncertainty of the context, from 0.2 to 0.05 and below, the query certainty is gradually increasing, though the pattern of this increase depends on the accuracy of the premises and, even more so, on the areal proportion of the categories (solid lines correspond with $\omega = 0.8$, dashed lines - with $\omega = 0.5$, and dotted lines with $\omega = 0.2$). Significant increase in query accuracy with the decrease of context uncertainty is achieved only for small absolute values of the context. Other experiments showed that the increase in premises certainty results in a modest increase of query accuracy until $\omega = 0.5$, while with $\omega > 0.5$ the result does not depend on how accurately the premises are specified. Strategies aimed at narrowing the uncertainty of the premises would be most successful if their absolute values are relatively low.

CONCLUSION

This work investigated the determinacy approach to formal modeling and resolving complex spatial queries, in which both elementary categories, and relations between them, can be specified with a certain accuracy. We showed that by accumulating the descriptions of relations between map layers as simple areal proportions, and identifying appropriate reasoning chains, it is possible to arrive at acceptable query accuracy without performing costly overlays. Query accuracy depends both upon uncertainty associated with categories and relations, and upon the absolute values of accuracy of the relations and the

context. Thus, such formal modeling can inform the user what elements of a query need re-specification should the user require a higher accuracy.

BIBILIOGRAPHY

Burrough, P. A. (1989) Fuzzy mathematical methods for soil survey and land evaluation. *Journal of Soil Sciences*, 40: 477-492.

Chesnokov, S. V. (1984) Sillogizmy v Determinacionnom analize (Syllogisms in Determinacy analysis). *Izvestiya Akademii Nauk SSSR. Seria Tekhnicheskaia Kibernetika*, 5: 55-83 (in Russian).

Chesnokov, S. V. (1990) Determinacionnaya dvuznachnaya sillogistika (Determinacy binary syllogistics). *Izvestia Akademii Nauk SSSR. Seria Tekhnicheskaia Kibernetika*, 5: 3-21 (in Russian).

Egenhofer, M. J. (1992) Why not SQL! *International Journal of Geographic Information Systems*, 6(2): 71-85.

Egenhofer, M. J., and Franzoza, R. D. (1991) Point-set topological spatial relations. *International Journal of Geographic Information Systems*, 5: 161-174.

Final Report of the Accuracy Assessment Task Force (1994) California Assembly Bill AB 1580 (California Department of Forestry and Fire Protection. NCGIA, UCSB).

Heuvelink, G. B. M., and Burrough, P. A. (1993) Error propagation in cartographic modeling using Boolean logic and continuous classification. *International Journal of Geographic Information Systems*, 7: 231-246.

Kolmogorov, A. (1951) *Foundations of the Theory of Probability.* Chelsea, New York.

Langran, G. (1991) Producing answers to spatial questions. In: *Proceedings of the Tenth International Symposium on Computer-Assisted Cartography, AUTO-CARTO 10* (Baltimore, MD: March 25-28, 1991), pp. 133-147.

Mises von, R. (1957) *Probability, Statistics, and Truth.* Allen and Unwin, London.

Ooi, B. C. (1990) *Efficient Query Processing in Geographic Information Systems.* Springer-Verlag, Berlin.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, CA.

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

Zaslavsky, I. (1995) Logical Inference About Categorical Coverages in Multi-Layer GIS. Ph.D. dissertation. University of Washington, Seattle.

SPATIAL METAPHORS FOR VISUALIZING INFORMATION SPACES

André Skupin
National Center for Geographic Information and Analysis
Department of Geography, 105 Wilkeson Quadrangle
State University of New York at Buffalo, Buffalo, NY 14261
voice: (716) 645-2722 ext. 32
email: skupin@geog.buffalo.edu

Barbara P. Buttenfield
Department of Geography, Campus Box 260
University of Colorado, Boulder, CO 80309
voice: (303) 492-3618
email: babs@colorado.edu

The growing volume and complexity of the World Wide Web creates a need for new forms of interaction with information. Spatial metaphors have been in the focus of interface research for a number of years. Recently, a related concept called *spatialization* has emerged as one possible strategy for dealing with modern information glut. However, the term remains ill-defined. We present a definition of spatialization that is based on the notion of *information spaces* that are non-spatial and high-dimensional. Through spatialization, they are projected into a low-dimensional form and made accessible for visual interpretation.

We implement this method to a body of about 100 newspaper articles. Following the extraction of keywords for each article, a multi-step process is applied. It involves the construction of a vector-space model, the computation of a proximity matrix and the projection into two dimensions via multidimensional scaling. The resulting coordinate configuration is imported into ArcView and linked with the keyword list. A number of visualization examples are shown, all based on a representation of each article as a point. One goal of this research is to investigate the feasibility of applying cartographic expertise to spatialized representations. Cartographic generalization is among the tools that could provide valuable inspiration for the visualization of large information spaces.

INFORMATION SPACES

Information is that which is inherent in a set of facts (Oxford Dictionary, 1996). An information space provides a well-defined strategy for organizing information. It can be formalized by logic, mnemonics, metric or nonmetric coordinates. The goal of the strategy is to facilitate navigation, browsing and

retrieval of items. One creates a structure to support access to the content, in effect.

Information spaces can be distinguished by the ways in which structure and content are interwoven into a specific physical and conceptual form. They are by no means new artifacts, introduced by late-20th century technology. Instead we have been surrounded by and interacted with information spaces for a long time. Newspapers are good examples. They contain chunks of information that is neatly organized into articles that are placed in physically defined locations on a page. To those that are familiar with the layout of a specific newspaper, it is an easy task to find and retrieve articles dealing with a certain topic, for instance the latest sports scores or developments in local politics. Given its relatively small volume, familiar organizational scheme, and physical nature, a conventional newspaper information space is relatively easy to navigate.

Other information spaces are more difficult to navigate. This may be due to the sheer volume of contained information, the non-physical nature of the storage or browsing medium, or to novel ways in which content is structured. Perfect examples are large hypermedia spaces, such as the World Wide Web.

SPATIALIZATION

Definition

In recent years it has been realized that new kinds of information spaces will require new methods for access. Among the most discussed strategic tools is the employment of spatial metaphors. Spatial metaphors are at the heart of a concept called *spatialization*. That term is applied in a variety of contexts, notably in digital audio processing where spatialization facilitates the identification of the location of sound sources in three-dimensional space.

Lakoff's (1980) use of the term spatialization is the most influential as far as the application of spatial metaphors in user-interface research is concerned. Kuhn (1992, 1996) introduced "spatialization" into the GIS interface jargon. Nevertheless, the term "spatialization" remains ill-defined. One common tendency is to use it synonymously with "the application of spatial metaphors". Spatialization can be defined more rigorously and literally, by establishing that formal spatial characteristics of distance, direction, arrangement, and pattern have or have not been achieved.

We define spatialization as

> a projection of elements of a high-dimensional information space into a low-dimensional, potentially experiential, representational space.

Information spaces are generally high-dimensional, given the complex, multifaceted character of their contents. Since the goal of spatialization is the creation of a cognizable representation, the latter has to involve fewer, typically two or three dimensions. It appears appropriate to use the term projection for the occurring transformation. Spatialization applies formal criteria to project a

view of contents into a reduced or simplified arrangement. Spatial metaphors provide natural strategies for orientation and navigation. Other aspects of spatial relations should also be established. These might include (for example) ascertaining that distances are commutative, that spatial autocorrelation applies to contents, or that changes in scale increase the level of apparent detail, in the spatialized solution.

Related Research

Hypertext and hypermedia have long espoused the idea of supporting navigation and retrieval through graphical representation of their structure and content. These representations have gained renewed attention with the advent and continuing growth of the World Wide Web. The majority of these visual representations is two-dimensional. Traditionally, they have been called *maps*. There are several principle ways in which visual representations of hypermedia spaces can be created. Some are merely manually created two-dimensional bookmark maps. Others, and those are the most common ones, are based on *structural characteristics* of the hypermedia space (Woods 1995, Mukherjea & Foley 1995). A third approach derives low-dimensional visualizations based on an analysis of the textual content of hypermedia spaces. It is mainly this approach that is being addressed by this paper.

Efforts are now being made to unify Web visualization with a more general file space visualization, exemplified by Apple's HotSauce, based on the Meta-Content Format (MCF).

In recent years much research effort has been invested into the investigation of spatial metaphors for user interfaces (Mark 1992, Dieberger 1994, Kuhn & Blumenthal 1996). Much of this is relevant and related to our notion of spatialization. Refer to Kuhn & Blumenthal (1996) for an interesting overview of the subject in tutorial form.

Surveying Information Spaces

In order to meaningfully spatialize information it has to be broken down into meaningful units or 'chunks'. This can be illustrated by evoking the image of a topographic surveyor who chooses to measure those surface points that have geometric or semantic relevance.

What are the units into which information can be divided? Some information spaces might appear to have a structure with an inherent "sampling unit". Examples for natural sampling units could be chapters of a book, single Web pages or newspaper articles. All these are, however, meaningful only at a defined level of interest. For example, there are instances when the focus is on a whole web site instead of a single web page. One might want to compare and correlate all the books on the shelf rather than all the chapters of a single book. What we are dealing with is the concept of scale. Like the surveyor, we have to consider both the intended scale and the purpose of our future representations in choosing meaningful sampling units.

As mentioned before, the goal of spatialization is to project contents of an information space into an easily cognizable representational space. In order to make the process consistent, its criteria have to be well-defined. One important aspect to consider is that elements of information spaces can be related to each other in many different ways. For instance, web sites can be related through such factors as content, connectivity, lineage, or geographical location. The combination of these factors forms certain configurations in an high-dimensional information space. It is the assumption of the spatialization approach that the metric qualities of these configurations can be numerically expressed and projected into a low-dimensional geometric space.

SPATIALIZATION OF A NEWSPAPER INFORMATION SPACE

Sampling the Information Space

Two editions of the New York Times, dated November 7 and November 8 1995, were chosen as input. We applied the spatialization concept to the 96 articles contained in Section "A". That time was just after the assassination of Israeli Prime Minister Yitzhak Rabin and the news contained a large number of articles highlighting various aspects of that event. Prominence of these current events should be reflected in the final information space.

As sketched above, the choice of a sampling unit is one of the most critical early decisions. The single newspaper article appears to be the best candidate, considering its coherency and its limited size, relative to the newspaper as a whole. Some other choices, like a division of the newspaper into pages with even or odd page numbers, would appear quite meaningless. However, in a sufficiently large newspaper, it might make perfect sense to divide its content into pages, each of which is dedicated to a certain subject area, like "Baseball", "Football", or "Hockey".

Next, a criterion must be chosen to distinguish articles from each other. Theoretically, there are again many choices. One could refer to an article according to its physical location in the newspaper, e.g. "Page 6, upper right corner". In fact, such a scheme can be useful when locating updated articles within well-known structures. For instance, in a certain newspaper the baseball scores might always be found in a certain location. Other factors could include the length of an article or the number of photographs associated with it.

Our approach assumes that the newspaper information space is only a special case of a much larger group of information spaces, including the WWW, to which the chosen method should be applicable. It becomes obvious that one factor will take precedence, namely, the content of the articles. It makes indeed little sense to compare "page 6, upper right corner" with "http://www....", but a comparison of their actual content can bear useful results. The content of articles forms the basis for our spatialization.

Technical Concept

The spatialization performs a content-based projection from the newspaper information space into a map space. This requires the definition of two major factors:

(1) Configuration of articles in the information space

(2) Projection method

Configuration of articles in the information space. This refers to the location occupied by each article in the high-dimensional information space. Following the principles of vector-space modeling (Salton 1989), this idea can be taken quite literally. In a vector-space model the occurrence of keywords in each article determines its location in an n-dimensional information space (n = total number of unique keywords).

If we assume that the chosen keywords sufficiently express the content of all articles, then the distance between articles in the n-dimensional information space is equivalent to their similarity. This is the assumption behind the widespread use of the vector-space model in many search engines, for instance on the World Wide Web. The principle is to use one or more search terms as input, form a vector of terms, and compare it to a stored list of vectors, each of which represents the contents of a web page. The resulting numerical values are the similarities/distances between the search term[s] and the web pages.

Projection Method. Tobler's first law of geography states that "everything is related to everything else, but near things are related more than distant things" (Tobler 1970). One of the primary assumptions of our approach is that a believable representation should be in accordance with that rule. Since the vector-space model already produced a configuration that expresses similarity through distance, a projection method is needed that strives to preserve these distance relationships.

One such method has been utilized by social scientists for many years: *Multidimensional Scaling* (MDS). It is a procedure that can be employed to transform a high-dimensional configuration, given in form of a proximity matrix, into low-dimensional coordinates. Over the course of several decades a variety of MDS algorithms were introduced and tested (Torgerson 1958, Kruskal 1964, Sammon 1969, Carroll & Chang 1970). The ALSCAL procedure (Takane, Young, and De Leeuw 1977) became the MDS method of choice for many statistical software packages, like SPSS and SAS.

Vector-Space Model. For each article a number of keywords was manually extracted. It was quite impossible to collect an equal number of keywords for each article, which varied in (a) the total length, i.e. word count, and (b) the level of generality. Some extremely short articles did not contain enough substance to produce more than five keywords. Other articles highlighted so many facets of a subject that even fifteen keywords hardly sufficed. On average, the essence of an article could be captured with the extraction of about ten keywords. Keyword extraction was performed independently by each co-author, and keyword sets compared to check for consistency.

A total of 415 unique keywords was extracted from 96 articles. They were merged to form a term vector T:

$$T = [t_1, t_2, \ldots, t_i] = ["AIDS", "advertisement", \ldots, "Rabin", \ldots, "Zyuganov"]$$

By matching vector T against each article individually, a term-article matrix (size 415x96) is created, with values of "1" indicating the presence of a keyword in an article and "0" indicating its absence. As a result, each article is characterized by a certain arrangement of "1" and "0" in one matrix column. That column vector describes the location of each article in the information space. The distance/dissimilarity of two articles can be computed by comparing column vectors. A variety of proximity coefficients exist to fulfill that purpose and the choice between them is somewhat arbitrary. After several tests, a Euclidean proximity measure was chosen:

$$\Delta_{jk} = \left[\sum_{i=1}^{n} (X_{ij} - X_{ik})^2 \right] \quad \text{(Sneath \& Sokal 1973)}$$

(n = 415; X = term-article matrix)

By applying this Euclidean measure to every pair of articles (n=415), a dissimilarity matrix is created (size 96 x 96). This matrix is input to the ALSCAL procedure in SPSS. The output is a two-dimensional configuration with coordinates for each article.

SPATIAL PRINCIPLES IN THE INFORMATION SPACE

Earlier in this paper, we argued that a rigorous spatialization must establish the presence of spatial metaphors such as distance, direction, arrangement and pattern. These four characteristics can be used to build up compound metaphors (autocorrelation, region building, intervisibility, etc.) The remainder of this paper will demonstrate two concepts, namely region definition and scale-dependence, in the newspaper information space. We begin with simple visualization and exploration.

Visualization

With the help of desktop mapping tools, the two-dimensional coordinates of each article can be linked with respective keywords. Figure 1 shows a point visualization in which a number of points have been labeled. Each point represents one article. The labels are created by accessing the first keyword identified for each article. Notice in the Figure that the keyword "Rabin" appears several times in the plot, thus we can determine that this visual display does not preserve spatial uniqueness: the same information "place" can appear in multiple locations. One might use this property to advantage, for example by linking a network between regions (article clusters, in the plot).

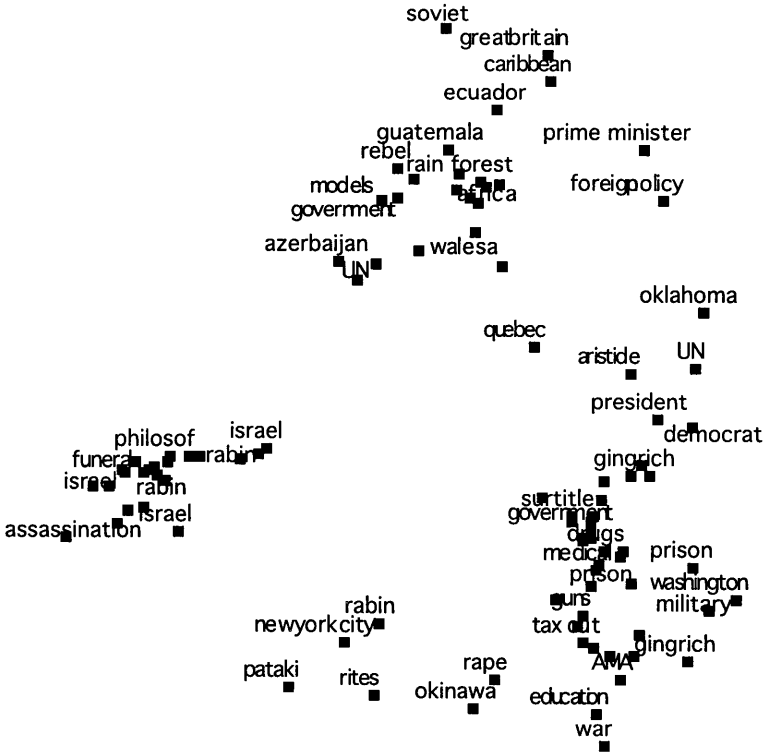


Figure 1. Spatialized Solution Represented in ArcView

One can perform simple queries that are common in desktop GIS. Figure 2a illustrates how the graphical selection of a number of articles corresponds to a highlighted section of the table. (The scatter plot in this Figure is a reduced version of Figure 1). In the selection rectangle, all articles refer to foreign events, from a U.S. perspective. In Figure 2b an attribute search is performed, searching for all articles with the keyword "assassination". Corresponding points are highlighted in the plot.

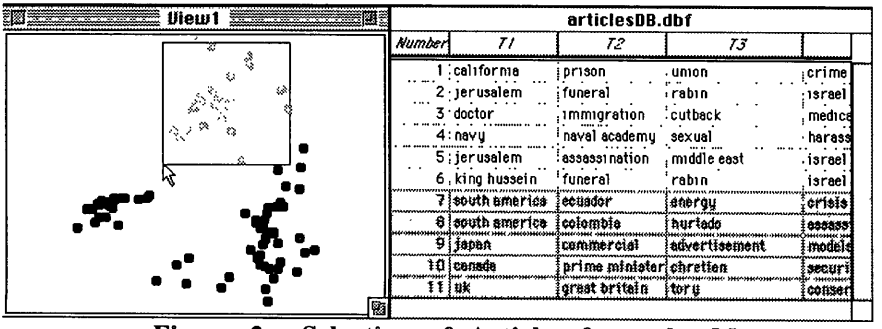


Figure 2a. Selection of Articles from the Map

These simple visualizations confirm three main clusters of articles: (1) domestic events in the lower right corner, (2) foreign events in the upper half, and (3) events in Israel on the left. The articles concerning the assassination and funeral of Yitzhak Rabin form a distinct and distant cluster. The exceptions are three points in the lower left of the map. These refer to articles surrounding Rabin's assassination that were related to the U.S.. Examples are the Jewish mourning in New York City and U.S. politicians attending the funeral in Jerusalem.

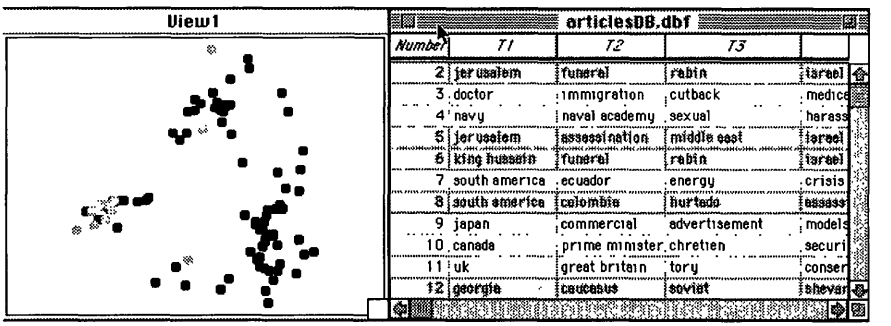


Figure 2b. Selection of Articles from the Table

Generalization and Scale-Dependence

As the focus of interest changes, e.g. from single books to single book shelves, so must the visual representation of information spaces change. Two options exist for obtaining more abstracted or more detailed representations.

One option is to initiate a new spatialization, with different sampling units. This would involve recomputing coordinates. Locations (and thus regions) in the new information space would not be comparable to those in the first. The other option involves a process that cartographers call generalization.

Its application to spatialized representations is most intriguing. Appropriate generalization permits exploration of the rate at which information densifies as scale changes, and will additionally preserve relative locations, permitting multi-scale analyses of the information space.

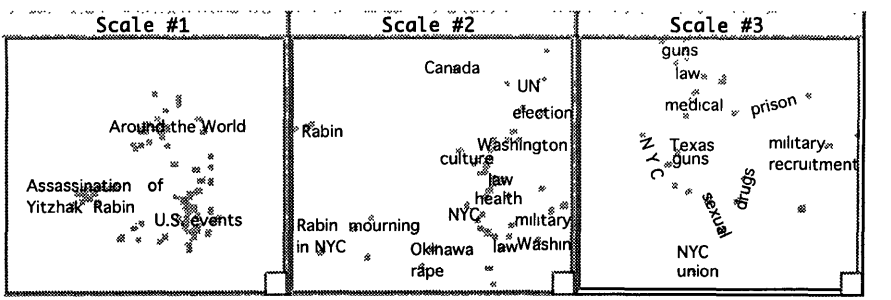


Figure 3. Visualization of spatialized data at three scales

Figure 3 shows a simple example of generalization applied to the New York Times information space. Three maps are shown, each at a different scale, with more specific content revealed as we zoom in. The center point is the cluster of U.S. events; as new keywords are resolved we can better define the fabric of the local information space.

SUMMARY AND SEARCH FOR MEANING

To the casual observer, the spatialization example may appear quite simple and almost trivial. One has to bear in mind that the complexity and sophistication of spatialized representations has to be in tune with the degree to which we can attribute meaning to each of the graphical components.

The processes leading up to the geometric configuration are complex. We must be careful to understand the mathematical and statistical assumptions underlying the geometry before reliable and meaningful interpretations of spatial relationships can be made. This paper is a 'proof-of-concept' demonstrating that existing statistical tools can be applied to generate information spaces, and that the presence or absence of simple spatial metaphors can be established to explore collections of information. As we move towards larger information collections, and towards more complex representations, one can envision three-dimensional models of information 'terrain analysis'. New questions might be asked, about the meaning of slope, intervisibility as a metaphor for indexing or cross-referencing. Other spatial analytic tools might be applied with varying degrees of effectiveness.

The spatialization of information spaces is an important application for geography. Early efforts for the mapping hypermedia structures were frustrated by the complexity of large hypermedia documents. In the late 1980's many hypermedia researchers even concluded that complexity stood in the way of navigating such documents and that spatial metaphors were unfeasible. What they ignored was that there existed a field of science and technology that had a wealth of experience in dealing with graphic complexity: cartography. Skupin & Wieshofer (1995) point out cases in which proven cartographic principles are being basically reinvented. Nielsen's (1990) ideas of "clustering" and "link inheritance" are examples. With the growth of the WWW, spatialized representations of hypermedia have found a renewed interest, but such notions as scale and region building remain virtually unknown in the hypermedia research community. This is a wide open field and cartographers have yet to discover it.

REFERENCES

Carroll, J.D., Chang, J.J., 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika, 35: 238-319.

Dieberger, A., 1994. Navigation in Textual Virtual Environments using a City Metaphor. Doctoral Thesis. Vienna University of Technology. Vienna, Austria.

Kruskal, J.B., 1964. Nonmetric Multidimensional Scaling. Psychometrika. 29(2): 115-129.

Kuhn, W., 1992. Paradigms of GIS Use. Proceedings 5th International Symposium on Spatial Data Handling, Charleston. IGU Commission on GIS.

Kuhn, W., Blumenthal, B., 1996. Spatialization: Spatial Metaphors for User Interfaces. Department of Geoinformation, Technical University Vienna.

Mark, D., 1992. Spatial Metaphors for Human-Computer Interaction. Spatial Data Handling. Proceedings 5th International Symposium on Spatial Data Handling, Charleston. IGU Commission on GIS.

Mukherjea, S., Foley, J.D. , 1995. Visualizing the World-Wide Web with the Navigational View Builder. Computer Networks and ISDN System, Special Issue on the Third International Conference on the World Wide Web '95, April 1995, Darmstadt, Germany.

Nielsen, J., 1990. Hypertext and Hypermedia. San Diego: Academic Press.

Salton, G., 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Publishing Company.

Sammon, J.W., 1969. A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers. C-18 (5): 401-409.

Skupin, A., Wieshofer, M., 1995. Cartography at the Hypermedia Frontier: Animation and Visualization of Hypertext Structures as two Examples. In: Mayer, F. (ed.) Wiener Schriften zur Geographie und Kartographie, Band 5.

Sneath, P., Sokal, R., 1973. Numerical Taxonomy. San Francisco: W. H. Freeman and Company.

Takane, Y., Young, F., De Leeuw, J., 1977. Nonmetric Individual Difference Scaling: An Alternating Least Squares Method with Optimal Scaling Features. Psychometrika, 42: 7-67.

Tobler, W. , 1970. A Computer Model Simulating Urban Growth in the Detroit Region. Economic Geography . 46 (2): 234-240.

Torgerson, W.S., 1958. Theory and Methods of Scaling. New York: John Wiley.

GIS ICON MAPS

Micha I. Pazner, Visiting Fellow
National Center for Geographic Information and Analysis
University of California Santa Barbara
USA

Melissa J. Lafreniere, Student
Department of Geography
The University of Western Ontario
Canada

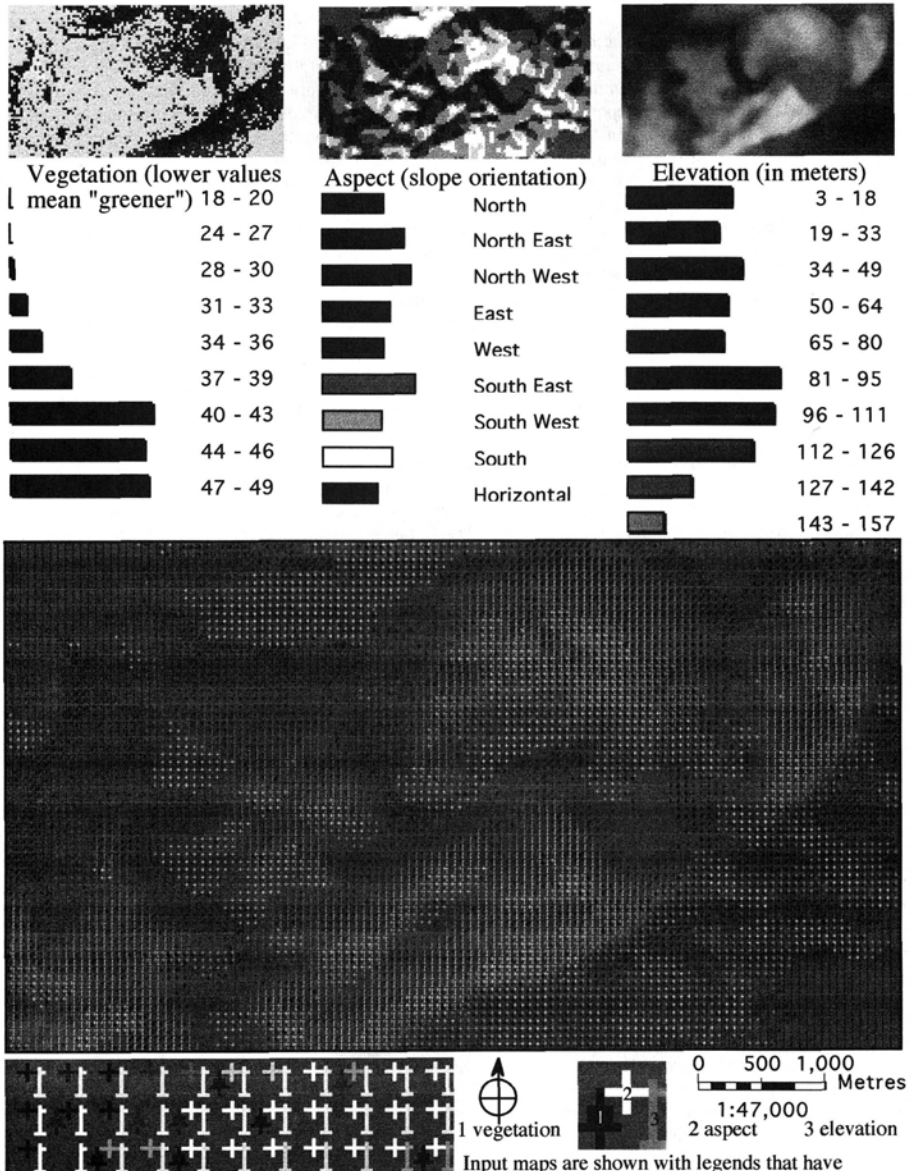
The paper presents and discusses image based GIS icon maps — a unique GIS visualization technique. A GIS icon construct allows us to map and record the interrelation of several cartographic variables at each location on a single image with minimal loss of information. GIS icon maps are designed for non-fused visualization of several variables. In these image-maps the basic assumption of each cell representing a single location is relaxed to permit multiple cells per original location. Every cell is transformed in the GIS icon map into a patterned matrix of output cells—the icon. While the underlying system's data structure remains single cell based, GIS icons can be seen as higher-level constructs that have sub-cell and super-cell properties. As a result, GIS icons are more graphically versatile than the conventional single pixel per location display. The only graphical element in a single pixel is tone or color. GIS icons support additional graphical elements including length, width, shape, angle and orientation. Examples of GIS icon maps that were created using terrain and reflectance remote sensing data are presented. This is followed by a brief discussion of the process for creating these maps. Careful design can lead to information-rich and aesthetic maps. The resultant maps reveal macro interaction patterns, while retaining in a zoomed-in micro view, full and easily visualized information for every variable at each location. The paper examines the GIS icon construct from a number of graphic and spatial science perspectives including computer graphics, spatial image processing in raster GIS, the elements of photo interpretation, issues in cartography, and E.R. Tufte's principles of graphic design in "Envisioning Information". It is concluded that GIS icon constructs can be used effectively for co-visualizing multivariate interrelation.

"At the heart of quantitative reasoning is a single question: Compared to what?" (Tufte, 1990).

GIS ICON MAPS—A VISUALIZATION TECHNIQUE

The research question is: How can dense pixel data layers be co-visualized — with each layer's information visible? This seemingly basic question is non trivial. It addresses the ongoing challenge of visualizing multivariate data. There are a number of image overlay techniques used in digital image processing. All involve tradeoffs that result in loss of information. GIS icon maps are unique in that this loss is minimized while the visualization objective is maximized. Figure 1 shows three input layers and one output icon map.

Image based GIS icon maps were developed as a soft copy and hard copy GIS (geographic information system) visualization technique. The stimuli for their development came from a number of different sources. GIS icon maps embody principles of graphic design based on Tufte's (1990) treatise on Envisioning Information. The methodological approach is an adaptation of the work on iconographic and glyph constructs for exploratory visualization in Computer Graphics (Erbacher, Gonthier and Levkowitz, 1995) (Levkowitz, 1991) (Levkowitz and Pickett, 1990) (Pickett and Grinstein, 1988) and (Pickett, Levkowitz and Seltzer, 1990) The software tools are those of spatial image processing operations using a raster GIS (Tomlin, 1990) (Pazner, 1995). The GIS icon is a result of adopting a combined approach involving principles of visualization, a computer science glyph methodology, and image based GIS tools.



Input maps are shown with legends that have identical gray sequences to those used in the icon map. The *North Arrow* also serves to point to the location of the enlarged section (Bottom Right). Note the relations between vegetation and aspect and elevation — and the patterns and textures that are created as a result in the icon map.

Figure 1: GIS Icon Map of Vegetation, Aspect and Elevation.

GIS icon constructs allow us to map and record, both visually and digitally, the interrelation of several cartographic variables at each location on a single image with minimal loss of information. Figure 2a presents a close up view of an icon map including the underlying digital values. GIS icon maps are designed for non-fused co-visualization of several variables. By non-fused it is meant that each original data variable is present and retains a separate visual and numeric identity. In other words, we are dealing here with a type of image overlay technique — one that is distinct in a couple of ways. First, the technique is distinct in its ability to provide an answer to the research question we have posed, i.e.: how can dense pixel data layers be co-visualized — with each layer's information visible? Conventional overlay methods, which apply overlay arithmetic and logic to a single pixel location at a time, involve loss of information as a result of data fusion or superimposition. Second, unlike most other overlay methods, this technique relies on a what may be termed a reworked data resolution in order to reach its goal. GIS icon maps, presented here (Figs. 1, 2, 4) use sub-cell resolution pixel blocks to generate spatial neighborhood overlay patterns within each location. In these image-maps the basic assumption of each cell representing a single location is relaxed to permit multiple cells per original location. Every cell is transformed in the GIS icon map into a patterned matrix of output cells—the icon.

EXAMPLES OF GIS ICON MAPS

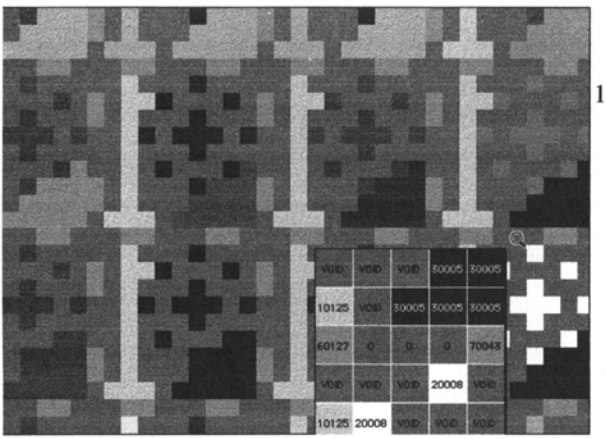
The examples shown here use remote sensing terrain and reflectance data from a SPOT satellite stereo pair. The study area, shown in Figure 3, is in the Campbell Hills, District of Mackenzie, NWT (Canada). Nine icon maps were created for terrain variables, reflectance variables, and mixed 'hybrid' icon maps that show the interrelation between terrain and reflectance variables (Table 1). The GIS icon maps were evaluated in terms of their usefulness for exploratory environmental visualization. Figures 2 and 4, for example, show an icon map of three terrain variables: Elevation (DEM) data and two of its derivatives: slope Steepness and Aspect. The DEM data was acquired from a SPOT satellite image stereo-pair. Additional terrain variables which were derived and used in other icon maps include slope Inflection and Drainage. Slope inflection is a measure of the amount of concavity or convexity of a slope location. Drainage presents a computed drainage pattern based on pouring 'digital rain' on the terrain.

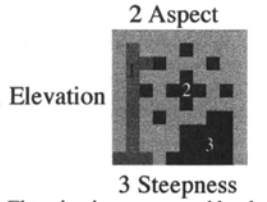
Examples of derived reflectance variables include the thematic results of remote sensing digital image classification. The reflectance variables were derived from three bands of SPOT data: Green, Red and Near Infrared. A 'V.I.S.' classification approach (Card, 1992) was used to derive the three land classes **V**egetation, **I**mpervious, **S**oil, and a Water class. The DEM data and the three bands of SPOT data are orthorectified, registered to a topographic map, and aligned to one another. Table 1 itemizes nine GIS icon maps by the terrain and reflectance variables used to produce them. Two of these are terrain variable maps (1, 2) one of which is shown in Fig. 4, two are reflectance variable maps (8, 9) and the five middle maps (3, 4, 5, 6, 7) are hybrids (for example see Fig. 1). Our experiments indicate that **3** is a good number of variables to use in GIS icon maps in terms of their construction, visualization and interpretation. As can be seen in Table 1, all but one of the maps are tri-variate. The hypothetical number of possible variables in GIS icon maps is $2 \rightarrow n$.

A good example of a GIS icon map is GIS Icon Map # 3 (in Table 1) which contains information on Vegetation, Elevation and Aspect (Fig. 1). This map presents meaningful information in a visually accessible manner. The interplay between the relationship of vegetation and aspect, and the relationship of vegetation and elevation (acting here as a surrogate variable for lithology) is evident in the interesting patterns and textures that appear on the icon map. The vegetation exhibits a clear preference for southerly illumination and for one of the lithologic units. A visualization of two variables exerting control over a third. The individual data layers are shown as separate maps above the icon map.

ESSENTIALS IN PRODUCING GIS ICON MAPS

We define a *GIS icon map* as a map in which every GIS pixel is transformed into a patterned matrix of output pixels—an icon. A detailed account of how GIS icon maps are produced is the focus of a previous





Elevation is represented by the 'flagpole'. Higher values are brighter and lower elevations are darker.

Aspect is represented by a 'sun'. Southerly facing slopes are brighter. North slopes are dark.

Steepness is represented by the 'stepped slope'. Steeper values are brighter.

Figure 2a: Close up view of a GIS Icon Map of elevation, steepness, and aspect. The screen shot shows the use of a numeric magnifying glass software tool to reveal the digital values (e.g.: "exactly how high?"..etc.) of each variable, including 'hidden' variables in the icon border.

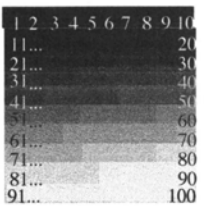


Figure 2b: Icon Addressing System.

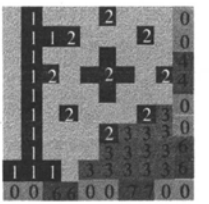


Figure 2c: The Icon Design.

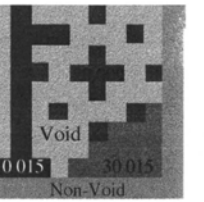


Figure 2d: Icon. loaded with data values

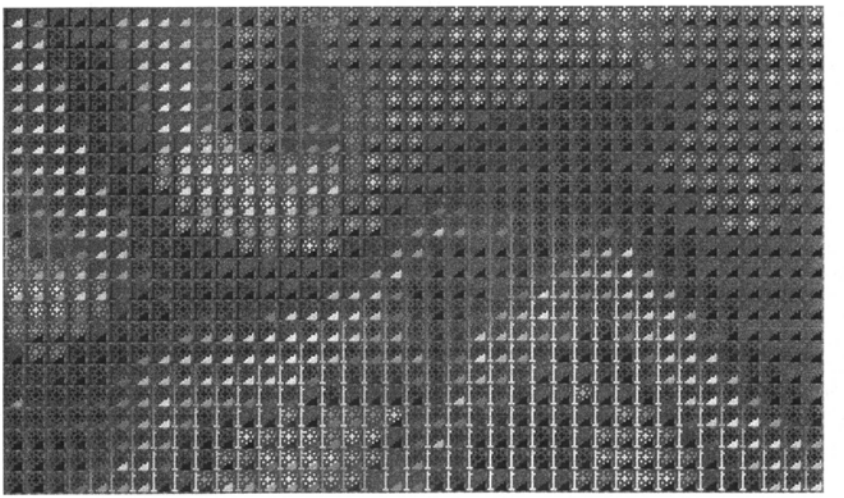


Figure 2e: Portion of a GIS Icon Map of elevation, steepness, and aspect.

article (Lafreniere, Pazner and Mateo, 1996). A very brief description of the process is provided here. GIS icon map experiments were done with Map·Factory (Limp, 1996), a raster GIS. A 10 by 10 matrix is used as a uniform project icon size in this study Icon maps with 3 variables were found to be potentially interesting, non-confounding and effective. A 10X10 icon provides sufficient space to design and place a variety of variable symbols, such as pictorial representations and alpha-numerics. A 9X9 area is used for the icon variables; one column and one row are reserved for creating a shared separating border around the icons (figs. 2c, 2d). A 10X10 icon means that the GIS icon map will contain one hundred times more image pixels than its input maps. There may be a need to reduce the size of the input prior to generating the icon map. This can be achieved by extracting a subscene, by sampling every nth row and column, or by interpolating to a coarser grid. Due to the trade-offs, several methods may be used in parallel. The input data is then rescaled to a finer cell resolution — in our case a fractional cell resolution of 0.1.

A prerequisite for creating GIS icon maps, *an icon matrix addressing system* is a relative reference system that is internal to the icon matrix and applied globally to all the icon matrices in the map (Fig. 2b). A key procedure generates the *icon matrix address map* with a cell resolution equivalent to that of the exploded map, where the cells in each 10X10 icon matrix are numbered sequentially from 1 to 100 (Fig. 2b). The value for a given cell reflects the relative position of that cell (the row and column coordinates) within the icon matrix . The purpose of the icon matrix addressing system is to enable the user to access and assign a new numeric label to a single location, or set of locations, within each address block in the address map (Fig. 2c). The procedure for creating the icon matrix addressing system is explained in detail in Lafreniere, Pazner and Mateo (1996). The addressing system is created using a logical set combination operation, with importance to order, of a cyclical row number map with a cyclical column number map, both cycling with the desired periodicity. Such row and column maps can be created using standard raster GIS operations that include distance measurement, automatic renumbering or category density slicing, and overlay subtraction.

Now that each icon matrix has a common addressing system, the variable symbol designs can be specified. The *icon design* is the result of assigning a new numeric label to a single location, or set of locations in the icon for each of the variables in the GIS icon image (Fig. 2c). Careful attention should be given to the graphic design of the variable symbol and its implementation as a set of pixels. The effectiveness and impact of the icon map will depend on the design of the variable symbols, the data, and its colorization. Once the icon design is determined the next step is to create a value template for each variable in the icon design. A *value template* consists of a map where the collection of cells that form the variable symbol have been loaded with the data values for the corresponding variable (Fig. 2d). The process for creating these maps involves three steps: creating the variable masks, processing the variable maps, and creating the value template for each variable. Unless the variables have been normalized, arithmetic adjustments need to be applied to the value templates to ensure that each variable has a unique set of values which can then be colorized. All of the cells within the icon matrix which do not represent variable values are assigned a null value. Unused border cells can be assigned a null or zero value. The final processing step is a straightforward overlaying of the value templates. The resulting map is the GIS icon map.

At this stage of the process the usefulness of the GIS icon map depends on our ability to colorize the results effectively. The map needs to be assigned tone and color sequences that have been carefully chosen for a specific visualization goal (Fig. 4). A gray-tone version is useful for getting a good first look and for generating non-color hardcopy output (Figs. 1, 2). In addition, a gray monochromatic scheme is also useful for image interpretation of the results; providing an equal and controllable good dynamic range of tones for each of the variables (Figs. 1, 2). At certain viewing scales, gray icon maps can take on a textured appearance (Fig. 1). Using color is tricky but can yield valuable visual patterns (Fig. 4). A single color sequence can be applied in many different ways to a particular variable, highlighting different information. Variables may be classed, ie. placed in value groupings, and the color sequences applied to the classes. Different color sequences assigned to the various variables will interact with each other and with the background colors (Fig. 4). It is important to note that the icon design strongly affects the visual patterning of the GIS icon image. For example, if one variable occupies a large area within the icon, this variable will tend to dominate the visualization. Therefore, icon design and colorization are critical factors

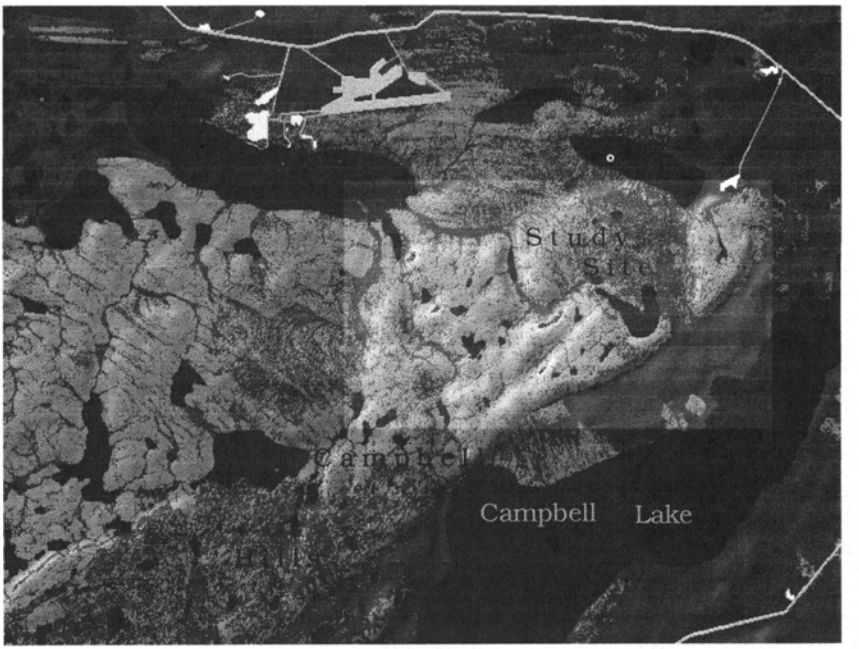


Figure 3: The Study Area in the District of Mackenzie, NWT (Canada). The study site in the Campbell Hills for which GIS icon maps were created is highlighted. The airport of Inuvik and the Dempster Highway are visible in this shaded relief, hydrology and vegetation satellite data composite.

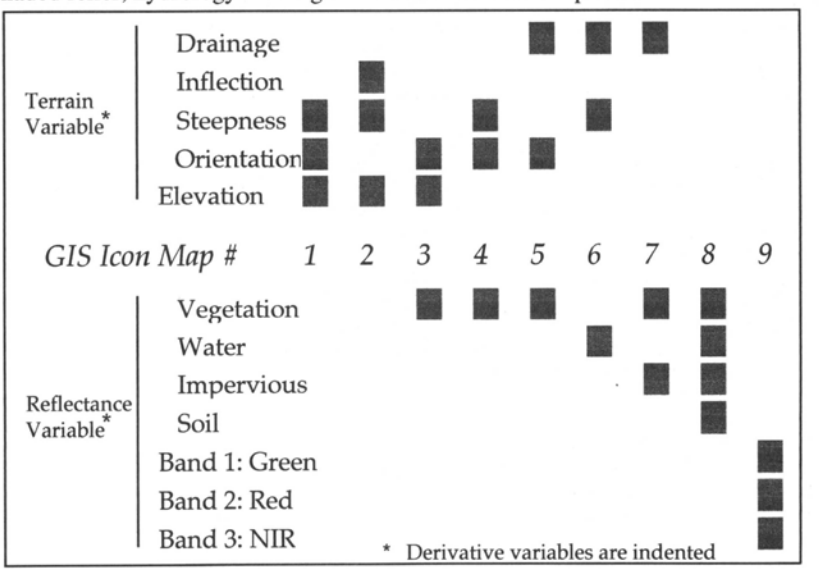


Table 1: Nine GIS Icon Maps of terrain and reflectance variables.

in the effectiveness of GIS icon maps. Good data, variable selection, icon design and visualization can lead to good interpretations.

EVALUATING ICON IMAGES FROM GRAPHIC AND SPATIAL SCIENCE PERSPECTIVES

GIS icon maps represent a visualization technique. This section examines the GIS icon construct from a number of perspectives including computer graphics, raster GIS, photo interpretation, cartography, and graphic design. GIS icon maps were inspired by the glyph and icon approach implemented in the Exvis system developed by the computer graphics group at the University of Massachusetts at Lowell (Erbacher, Gonthier and Levkowitz, 1995) (Levkowitz, 1991) (Levkowitz and Pickett, 1990) (Pickett and Grinstein, 1988) and (Pickett, Levkowitz and Seltzer, 1990). An example and brief explanation of Exvis appears in a chapter on multivariate geographic displays (DiBiase et al., 1994) in a book on visualization in modern cartography, and in the book Visual Cues (Keller and Keller, 1993). In Exvis the icon design is based on pixelated line segments somewhat similar to a stick figure with limbs portruding at various angles. Each line segment represents a variable. The strength of the variable determines the orientation (angle) of the line segment. The macro result are 'velcro maps' or images exhibiting various surface textures. A sound (acoustic) interface helps interpret regional combinations. Due to the relatively large number of variables represented (e.g.: five, seven, etc.) and the uniform graphic design of each variable as a line segment, it is difficult to interpret local features in the image. As such, one could argue that the multivariate visualization product ends up graphically scrambling and hiding information on how the variables are co-related.

However, the fundamental notion of an icon design where an array of pixels is used to represent each original image location, is sound and holds considerable potential — given proper implementation. Our question was: is it possible to develop an icon image variant in a raster GIS environment? As this paper and Lafreniere et al. (1996) demonstrate, it is indeed possible to develop a raster GIS model, or procedure, that leads to the derivation of GIS icon maps. The model is implemented as a macro-like script which can be reused, altered and adapted to varying data, icon sizes, and icon element designs. A key difference from Exvis is that GIS icon design is based on fixed micro location variable templates, with color used as the graphic means to portray variability. There are advantages to this static design for the end user in terms of interpreting clearcut results based on a non-varying geometry. And it gives the map-maker control over the variable template design (Fig. 2c), allowing him/her to create fixed diverse pictorial cues (e.g. a sun template to reflect illumination, a tree template to represent vegetation, etc.). It also gives the map-maker control over physical separation and graphic differentiation between variable templates in the icon design stage.

GIS icon design has a special ability to take into account the classic elements of photointerpretation (Avery and Berlin, 1992). Visual variables such as color, length, width, orientation, shape, size, pattern and texture can be designed and implemented for optimal visualization impact (Buttenfield, 1993) (Bertin, 1983). While computer assisted image interpretation is routinely used to create derivatives for the human interpreter, it has been generally limited to affecting pixel tone/color. A good example are false color composites of remote sensing data which are single pixel column based. Both color composites and icon maps are multivariate visualization products, and tend to be well suited for use with three variables. They differ in that only icon maps offer multi-pixel, non-fused, and patterned co-visualization of the variables. A multi cell approach is needed in order to represent and manipulate higher order elements of photointerpretation than tone/color. While the underlying system's data structure remains single cell based, GIS icons can be seen as higher-level constructs that have sub-cell and super-cell properties. As a result, GIS icons are more graphically versatile than the conventional single pixel per location display. The icon map approach points to the fact that there are substantial advantages to moving from a single pixel processing mode to an aggregate pixel block (such as the icon). The image-processing and map-making analyst that prepares GIS icon maps has some control over designing elements of photointerpretation that can then be used by an image interpreter — the end user — which may or may not be the same person.

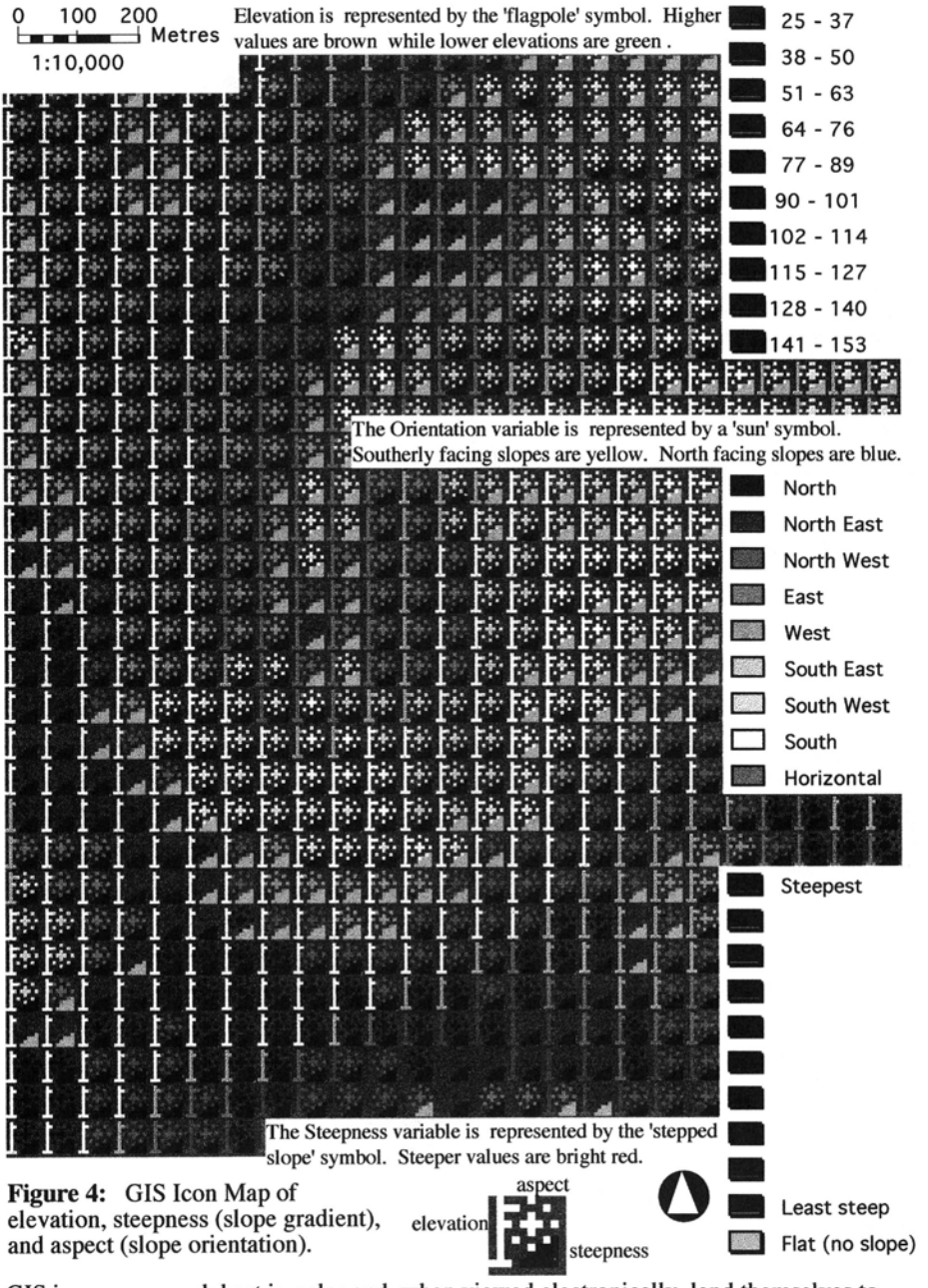


Figure 4: GIS Icon Map of elevation, steepness (slope gradient), and aspect (slope orientation).

GIS icon maps work best in color and, when viewed electronically, lend themselves to dynamic data exploration of multivariate interrelation. Macro views reveal general trends and patterns, while micro views stimulate local interpretations and hypotheses.

GIS icon maps represent an interesting cross between a modern pixel-based image and a traditional map composed of cartographic symbols. This has several implications on their use. Analyzing an icon map may require a type of hybrid map reading and image interpretation skill — which may take some getting used to. On the other hand, the potential exists to reap the advantages of interpreting image data while reading easily recognizable cartographic symbols. In reading icon maps such as Figs. 1 and 2e, the emphasis is on comparing between the 3-4 variables that are depicted on each map. This is an exercise in interpreting isolated-and-joined elements that is different from the interpretation of normal imagery (Fig. 3). The interpretation of icon maps involves explicit comparison of the inter-relation of the mapped variables.

Principles of graphic design based on Tufte's (1990) treatise on Envisioning Information were deliberately incorporated into the design of GIS icon maps. Tufte discusses issues in *layering and separation* in his book. Icon maps are inherently layered and can be read knowing that none of the layered information is neither covered nor fused (Fig. 1). GIS icon maps are also a good example of what Tufte terms *micro-macro readings*. Consequently these graphic representations can be read at a continuum of scales from fine micro detail, through meso, to macro levels of detail (Figs 1, 2). Another well known design that Tufte advocates using is *small multiples* (e.g. the input maps at the top of Fig. 1). As this study shows (Table 1), it makes sense to design a family of icon maps, each depicting a subset of 3-4 variables. A graphic layout of a number of such maps would be a good example of a small multiples graphic design. GIS icon maps provide non-fused co-visualization within the eyespan. The visually continuous and uninterrupted presentation of information within the eyespan runs as a recommended common design thread in Tufte's exposition of graphic practises. Many of the guidelines for the use of color provided by Tufte in his chapter on *color and information* can be readily applied to colorization of icon maps. This should come as no surprise since Tufte draws heavily on Eduard Imhof's (1982) rules for the use of color in cartography. Examples of color principles that are readily applicable to icon maps include the use of muted background colors, the use of colors found in nature, the sparing use of very strong colors and contrast, and the need to apply damage control measures to mitigate negative effects of interacting color elements. A color example is shown in Figure 4.

The GIS icon map is a result of implementing a computer graphics glyph method in an image based GIS environment while incorporating a set of graphic design principles. The software tools are those of spatial image processing operations using a raster GIS (Tomlin, 1990) (Pazner, 1995). The model is unique in that it approaches the map overlay problem in GIS in a non-standard way, based on changing the cell resolution and using cell aggregates to achieve a spatial neighborhood based overlay. From a modeling standpoint, the process of creating GIS icon maps can be seen as exploratory visual modeling or exploratory data visualization. Similar to exploratory data analysis, the process, not just the result, constitutes an important part of the exploration. Exploratory visual modeling is achieved by performing various modeling steps: the selection of sets of three variables, the design of the icons, processing the data derivatives, running the GIS icon map generating model, coloring the results, and visual interpretation of the results. The procedure can be done by a sole researcher or a team, and leads to familiarization with the interrelation of data variables which in turn stimulates interpretations, hypotheses, and new research questions. Augmenting rather than replacing conventional one-pixel-per-location maps, GIS icon maps can serve a unique and useful role in visualizing the interrelation of multivariate data Possible applications include visualization of natural and artificial spatially distributed variables, gradients, indices, and uncertainty. The relative simplicity and flexibility of the GIS icon map technique makes it a powerful tool for non-fused visualization of multivariate data. Interpretation of icon maps can suggest further GIS modeling in order to derive additional quantitative and visual results. With proper graphic design, GIS icon maps can be created that have substantial aesthetic appeal.

ACKNOWLEDGMENTS

NATO Defence Research Group, Panel 8 (VR for training), Research Study Group RSG.30. The National Center for Geographic Information and Analysis (NCGIA) at the University of California Santa Barbara.

REFERENCES

Avery T. E., and G. L. Berlin (1992). *Fundamentals of Remote Sensing and Airphoto Interpretation.* Macmillan Publishing Company, New York, Fifth Edition, 476p.

Buttenfield B. P. (1993). Scientific Visualization for Environmental Modeling: Interactive and Proactive Graphics. 11p. *Proceedings of the Second International Conference/Workshop on Integrating Geographic Information Systems and Envrionmental Modeling,* NCGIA, USA.

Bertin J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps.* University of Wisconsin Press, Madison, WI. Translated by William J. Berg.

Card D. H. (1992). Characterizing Urban Morphology Using Remote Sensing and VIS Modeling. *1992 AAG Annual Meeting Abstracts Vol.* Assoc. Amer. Geographers. USA. p 33.

DiBiase D., C. Reeves, A. M. MacEachren, M. vin Wyss, J. B. Krygier, J. L. Sloan and M. C. Detweiler (1994). Multivariate Display of Geographic Data: Applications in Earth Systems Science, chapter 15 in *visualization in modern cartography,* A. L. MacEachren and D. R. F. Taylor, Eds. Elsevier Science Ltd U.K , pp. 292-293.

Erbacher R., Gonthier D., and H. Levkowitz (1995). The color icon: A new design and a parallel implementation. *SPIE,* Vol. 2410, pp. 3030-312.

Imhof E. (1982). *Cartographic Relief Presentation,* edited and translated by H. J. Steward from Imhof's Kartographische Gelandedarstellung (Berlin 1965). Berlin.

Keller P. R. and M. M. Keller (1993). *Visual Cues.* IEEE Computer Society Press and IEEE Press, USA, 229p.

Levkowitz H. Oct. (1991). Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters. *Visualization 1991,* CA:IEEE Computer Society Press, pp. 22-25.

Levkowitz H. and R. M. Pickett (1990). Iconographic integrated displays of multiparameter spatial distributions. In B. E. Rogowitz and J. P. Allebach, eds, *SPIE '90,* Human Vision and Electronic Imaging: Models, Methods and Applications, Santa Clara, CA, Feb. 12-14, pp. 345-355.

Limp F. (1996). Map•Factory 1.02. Software review, in *GIS World,* July 1996, GIS World Inc., Ft. Collins, CO, pp. 86-87.

Pazner M (1995). Cartographic Image Procesing With GIS. *GEOMATICA,* Canadian Institute of Geomatics, Ottawa, 49 (1): 37-48.

Pickett R M and G. G. Grinstein (1988). Iconographic Displays for Visualizing Multidimensional Data. *1988 IEEE Conf. on Systems, Man,and Cybernetics,* Beijing and Shenyang, People's Republic of China.

Pickett R., H. Levkowitz and S. Seltzer May (1990). Iconographic displays of multiparameter and multimodality images. *First Conference on Visualization in Biomedical Computing,* Atlanta, GA:IEEE Computer Society Press, pp. 58-65.

Tufte E R. (1990) *Envisioning Information,* Graphics Press, New Haven, CN.

Tomlin C. D. (1990), *Geographic Information Systems and Cartographic Modeling.* Prentice Hall, New Jersey, 249p.

SPATIAL DATA MODELS IN CURRENT COMMERCIAL RDBMS

Matthew McGranaghan

Associate Professor
University of Hawaii
Geography Department
2424 Maile Way
Honolulu, HI 96822 USA
matt@hawaii.edu

ABSTRACT

Vendors of database software are adding spatial capabilities to their products. This paper compares the spatial data types and the spatial relations which are recognized by a set of current commercial RDBMS and layered spatial database management packages. While there is convergence on the notion that databases should handle spatial information, there are both gross and subtle differences in what this means. A range of types, relations, capability, and ease of use is evident.

INTRODUCTION

Developments in database technology enable change in the practice of automated cartography and GIS, but impose limits through the data models they support. This paper describes the spatial data models that several current vendors of extended relational database management systems implement. It compares the richness of the data types, relations, and operators that each of the several vendors now offer. Minimally, these need to support an applications needs. More ambitiously, they might be hoped to support interoperability of heterogeneous spatial databases. However, the range of differences, both in substance and in nomenclature may inhibit this.

The relational model (Codd 1970) and relational databases have proven extremely useful for many applications, but, only a few years ago, many in GIS thought that spatial data were poor candidates for processing in relational database management systems (RDBMS). More recently, it has become apparent that extensible relational and object-oriented databases can meld spatial and mainstream databases (Strickland 1994, Abel 1996). van Oosterom (1993)

describes three approaches to incorporating spatial data in a relational database. He refers to them as dual architecture, layered architecture, and integrated architecture. We will not consider dual architecture further.

The layered architecture approach provides a layer of abstraction, in practice a set of relational tables containing the spatial information, between the user and the kernel of a standard RDBMS, which allows the user to think in terms of a spatial model but then translates this into data types which are native to the RDBMS. van Oosterom cites System 9, GEOVIEW, and CSIRO-DBMS (Abel 1989) as examples. ESRI's SDE is another.

An integrated architecture is one in which the types of data in the database are extended to include spatial data types. van Oosterom sites Intergraph's TIGRIS, and several research systems, including his own GEO++ (van Oosterom and Vijlgrief 1991), based on Stonebraker's Postgress. Strickland (1994) suggests that this ability to extend the relational model has allowed it to subsume object-oriented database functions, and the resulting object-relational model of extended RDBMS seems poised to carry the current vendors forward.

Mainstream commercial RDBMS vendors have also begun to offer spatial data handling capabilities. However, the claim to have "spatial capabilities" is vague. Vendors have taken different approaches, developing different data types and recognizing different relationships among spatial entities in these extended models. This paper is an attempt to make a *pima facia* comparison of the spatial data models, spatial operators, and richness of the application programmer's interface (API) provided in each of the major RDBMS' current release, based on sales literature, product documentation, published descriptions, and inquiry. It does not reflect hands-on experience or experimentation with the software. It certainly may reflect misconceptions that more familiarity would correct. The objective is to illuminate differences and similarities among the systems' spatial capabilities. System administration, management, and security issues are ignored.

DESCRIPTIONS

The following sections describe the objects, relations, and functions in each product. The descriptions are grouped by system architecture.

Layered Architecture Systems

Some information was available regarding Sybase Spatial Query Server, ESRI's SDE, and CSIRO-SDM (following SIRO-DBMS), all of which use a layered architecture.

SIRO-DBMS (Abel 1989), implements spatial objects and an extended SQL interface that recognizes spatial relations among them. This tool-box is built as a lean layer upon which developers can build custom systems. The data model is very similar to the U.S. Spatial Data Transfer Standard (SDTS) data model. It includes: point, rectangle, simple line segment, line string, link, directed link, chain, complete chain, area chain, network chain, ring, and both simple and

complex polygon types). Topology is carried in relations of entities and components. Points can be represented in relational tables. More complex features can be built up from them, or have coordinates represented directly in BLOB fields. It implements eight spatial operators as database search qualifiers: mbrint, intersects, encloses, is_enclosed_by, crosses, is_connected_to, within_buffer, and adjacent_to. Both the spatial model and operations are very well chosen for GIS applications. They are accessed via a C API of approximately two dozen functions. Quadtree-based spatial indexing is used to enhance performance, and comparisons with Oracle's OMD indicate that SDM is faster (Zhou 1995).

Sybase has spatial facilities through Spatial Query Server, a product from Vision International, a division of Autometric Inc. The Spatial Query Server (SQS) supports the definition of spatial data types spatial operators, and a spatial indexing schema, within a Sybase SQL Server. The eleven supported Spatial Data Types (SDT) are: point, rectangle, circle, ellipse, azimuth, line, polygon gpolygon (polygons nested within another polygon), voxel, polygon_set, and rectangle_set. The spatial qualifier operators are: intersect, inside, outside, beyond, and within. Spatial queries are processed against templates defining a geometric shape (i.e., rectangle, ellipse, circle, point, line, polygon, or gpolygon). A spatial index may be declared on a column of an SQS data type.

ESRI's Spatial Data Engine (SDE) sits atop an Oracle (or other) RDBMS but builds its own spatial layer eschewing Oracle's spatial data facilities. Reputedly, for better performance. SDE's spatial types include: point, point-cluster, spaghetti line, non-self-crossing line string, ring, polygon, and donut polygon. Polygon nesting beyond a "doughnut hole" requires the definition of separate (perhaps also "doughnut") polygons on down to the least enclosed polygon. Spatial indexing is through a three-tiered partition of the user coordinate system set up at the time of database creation. The API consists of 138 functions to enter, manage, edit, query, and annotate spatial and related attribute data. The SE_RELATION() function generates a bit-mask indicating which of the following eight spatial relations hold among two objects (line intersection, point in common, common boundary in same order, common boundary in reversed order, spatially identical features, area intersect (at least one feature is an area and the other is at least partially inside it), primary feature contained by secondary feature, and secondary feature contained by the primary feature).

Integrated Architecture Systems

Oracle's Spatial Data Option, CA-OpenIngress' Object Management Extension (OME) and Spatial Object Library (SOL), and Informix's Illustra 2D and 3D Spatial DataBlades are examples of commercial integrated architecture spatial relational databases. Some information was available for each of these.

Oracle's Spatial Data Option (formerly Oracle7 MultiDimension) claims support for three types of objects (point, line, polygon) and three spatial relations in promotional literature but the reference manual describes a more complex two

tiered system of access. First, bins of data on retrieved based on five relations (enclose, enclosed-by, overlap, equal_to, outside) between bins and three types of query window (range, proximity, and polygon). Second, data points stand in one of three relations (inside, outside, on_the_border) of the query window, while line segments can also overlap the query window (Oracle 1996).

Oracle's basic spatial model is that of a point in an up-to-32-dimensional space. The HHCODE represents a linearly-indexed bounded cell in the multi-dimensional data space. The range query retrieves HHCODES falling in an n-dimensional minimum bounding rectangle. The proximity query returns HHCODES within a radius of a point, with the assumption that each dimension is scaled the same. While one suspects what the polygon query should return, it is not really clear. The opaqueness of "a polygon window is defined by specifying a start and end point for each node, in two dimensions, up to a maximum of 124 nodes" (Oracle 1996, p. 3-7) is daunting. Predictably, the API is more database- than space-oriented.

CA-OpenIngress with the Object Management Extension (OME) and Spatial Object Library (SOL) implements geographic data types and geometric SQL functions. SOL comes from a partnership with Mosaix Technologies Ltd, and "provides a rich set of library elements for application development. Data involving spatial relationships can be handled by the database in the same manner as the more traditional data types of characters and numbers. Using these spatial shapes and functions, location data can be integrated easily into business applications." Details on the relations and entities, however, are not provided in the literature that has been examined to date.

Illustra Information Technologies Inc. (since December 1995 a subsidiary of Informix Software Inc.) calls its Illustra Server a "dynamic content management system". The 2D Spatial DataBlade Module supports ten spatial types: circle, directed graph, ellipse, line segment, path, point, polygon, polygon set, quadrangle, and square/rectangle (Informix 1994, p. 2-1). Coordinates are double precision. (Paths are allowed to be self-crossing. Polygon sets allow nested and disjoint polygons via explicit parent-child enclosure declarations.) Four functions (insert, update, copy, and micopy) convert external (string) representations to internal C structure representations. The select function returns a string representations of objects. While promotional literature indicates that 2D Spatial DataBlade provides "over 200 functions" to create, compare, manipulate, and query spatial objects, this count is generated by overloading fifty-seven operators to handle multiple object-types as function arguments. For instance, the Boolean operator *overlap*, can be called to compare objects of any type(s). The operators return the range of standard as well as spatial data types.

The 3D Spatial DataBlade Module has eighteen 3D data types, (including point, box3d, quadrangle, circle, ellipse, line segment, path, polygon, polygon set, vector, unit vector, circular arc, rectangle, polyline/polyarc, polycurve, polygon mesh, polygon surface, and polyface mesh). While promotional materials claim "over 1,000" functions, these are in fact arranged as sixty-four

overloaded function calls. Key relations (location, distance, overlap) are incorporated in the database and accessible via SQL and a C programming API. R-TREE spatial indexing is used as are "smart objects" in which the system decides whether to store coordinates for small objects in a relation or as a "large object".

COMPARISON

The following table summarizes the counts of data types and spatial relations described above, attempting to penetrate marketing rhetoric. It appears that the number of spatial data types ranges from 2 up to 18. This would seem to reflect an ultimate foundation of point-based vector representation, coupled with some differences in the level of abstraction and integration with which the system is designed to work. The number of relations among objects in the data base appears to range between 3 and 8 (or ?). There is much more variation in the API sizes, which range from on the order of three up to 138. This range reflects more the intended uses of products and the number of spatial relations recognized in queries, and the number of functions in the application programmers interface (API) as indicators of richness or expressiveness of the representation and the system. The size of the API may also indicate how complex it is to use. The table shows that there are differences, but masks what they are.

System	Data Types	Relations	API size
Sybase SQS	11	5	?
ESRI SDE	7	8	138
SIRO-DBMS	12	8	~30
Oracle SDO	2	3	3
CA SOL	?	?	?
Informix 2D SDB	10	4	57
Informix 3D SDB	18	4	64

The following sections attempt to more precisely compare the features of these systems. The effort is made more difficult by differences in nomenclature and the amounts of information that were available for each system.

Data Types

The following table indicates, for comparison, the spatial data types in each system. It raises several difficulties with the information available from vendors. These include differences in nomenclature, such as differentiating among several meanings of "complex polygon", and differences in levels of generality. For instance, identifying a "quadrilateral", or for that matter a "rectangle" type in addition to the more general "polygon" in literature describing a system, without also indicating how and whether there really is specialization, obfuscates rather than clarifies system capabilities --- one might as usefully list pentagons,

hexagons, etc. as types. Because of the limited treatment of 3D data in most of these systems, and for the sake of brevity, the table leaves out many of the 3D types in 3D Spatial DataBlade. Exclusion of a type from this list does not mean that it can not be implemented in a given system, only that it is not mentioned as a type in the system's data model. In any event, entries in the table are my best guess at native types from the information I have available.

Types	SQS	SIRO-DBMS	SDE	SDO	2D-SDB	3D-SDB
Point	Y	Y	Y	Y	Y	Y
Point Cluster	N	N	Y	N	N	N
Line Segment	N	Y	Y	Y	Y	Y
polyline	N	Y	N	Y	Y	Y
Ring	N	Y	N	Y	Y	Y
Topological Arc	N	Y	N	N	N	N
Simple Polygon	Y	Y	Y	Y	Y	Y
Complex Polygon	Y	N	N	N	Y	Y
Donut Polygon	N	Y	Y	N	Y	Y
Nested Polygon	Y	N	N	N	Y	Y
Circle	Y	N	N	Y	Y	Y
Ellipse	Y	N	N	N	Y	Y
Rectangle	Y	Y	Y	N	Y	Y
Rectangle Set	Y	N	N	N	N	N
Quadrilateral	Y	N	N	N	Y	N
Graph Network	Y	N	Y	N	Y	N
Layer	N	N	Y	N	N	Y
Azimuth	Y	N	N	N	Y	Y
Voxel	Y	N	N	N	N	Y

Only SIRO-DBMS seems to consciously support the topological arc notion that is at the core of the US spatial data infrastructure. It is not clear whether this signals anything. Perhaps it signals a general pulling back from data models that have supported much analytic cartography and GIS. Perhaps it signals realization that similar information can be recovered in reasonable time from other types. Or perhaps it signals unfamiliarity with existing spatial data processing techniques and / or a continuing model of separation of database from spatial data processing.

Relations

Comparing these systems on the relations that they understand also proved elusive. All of the systems provide some ability to extract data based on spatial relations, essentially allowing spatial relations among objects to become part of the qualifying predicate in a database selection operation.

The table below indicates the relations that each system recognizes. It appears that these systems are very well matched in this regard, however, from the system documentation, one once again gets the sense that there are

differences in meaning among interpretation of these concepts. One difference is in the types of data returned by queries on a given relation.

Operation	SQS	SIRO-DBMS	SDE	SDO	2D-SDB	3D-SDB
MBR Intersect	N	Y	N	N	N	N
Object Intersect	Y	Y	Y	Y	Y	Y
Enclosure	Y	Y	Y	Y	Y	Y
Proximity/Buffer	N	Y	N	N	N	N
Contiguity	N	Y	Y	N	N	N
Spatial Equality	N	N	Y	N	Y	Y
Exclosure	Y	N	N	Y	N	N

The APIs and Operations

Comparing the number of operations that each system can perform on data to produce new information reveals a bit about system orientation. Several of the systems allow what one might consider more traditional GIS, or other application, capabilities than simple data retrieval so that in addition to treating relations as spatial qualifiers on retrieval, they can also return newly computed spatial objects or measurements that result from performing a spatial operation on qualifying objects. This capability makes a system seem more like a GIS programming environment than a database interface, and indicates further migration from a dual model of separating spatial and attribute data. Drawing a distinction between what is retrieval and what requires computation of new spatial objects or information in not easy. The following table attempts to characterize each system by the number of functions it offers, in categories structured after Roger Tomlinson's list of seventy-two GIS functions.

Function Class	SQS	SIRO-DBMS	SDE	SDO	2D-SDB	3D-SDB
Input, Edit, Convert	?	1	2	0	2	2
Overlay	?	1	3	1	3	3
Buffer/Corridor	?	1	1	0	0	0
Display	?	0	0	0	0	0
Elevation modeling	?	0	0	0	0	3
Other	?	7	14	5	14	13

From this point of view, SDE is very GIS-like at the outset and the 2D and 3D Spatial DataBlades have considerable analytic geometric capability. Oracle SDO and SIRO-DBMS both seem more oriented toward accessing data for use by an application program.

CONCLUSIONS

From the comparison, a few general conclusions can be drawn. Vendors have recognized that spatial data are worth supporting, and their attention to this is paves the way for more GIS and cartographic use of RDBMS. Data access will be increasingly easy for even very large databases.

The distinction between GIS and database software is blurred by the spatial capabilities of current RDBMS. These range from bare data item retrieval to fairly full analytic geometric manipulation of spatial data. Products with large libraries seem to provide many GIS facilities. Expansion in this direction seems likely. It is not yet clear what this will mean for GIS development. The potential for GIS to be subsumed within database technology in the not too distant future seems real, but systems to date lack much in the way of data input and spatial co-registration.

The ranges of variation among the data types and operations are noteworthy. While progress has been made toward standardization in the spatial data community, and there is considerable convergence among the models in these systems, the differences are impediments to adoption and to interoperability. It is not yet clear which of these sets of spatial types and relations will prove to be "most adequate". Public benchmarks showing performance on a suite of common GIS tasks under each data model would be interesting. So would demonstrations of the effort required to move between these data models.

The problems in trying to make information about these systems commensurate, and the concomitant limitations of the present paper should be recognized. Differences in nomenclature and nuance make description-based comparison suspect. This work was conducted without side-by-side access to test these products experimentally. Additionally, firms have considerable financial interest in the reputations of their products, and some consider detailed descriptions of their capabilities to be proprietary information. The cooperation of several firms in providing demonstrations and brief access, or even simply being willing to sell documentation beyond sales literature to non-licensees of their software is appreciated.

REFERENCES

Abel, D.J. (1989) SIRO-DBMS: a Database Tool Kit for Geographical Information Systems. *International Journal of Geographical Information Systems*, v3, n2, p103-116, 1989.

Abel, D.J. (1996) What's Special about Spatial?, *Proceedings of the 7th Australasian Database Conference*, Melbourne, Australia, January 29-30, 1996.

CA-OpenIngres, Object Management Extension (OME) and Spatial Object Library (SOL) sales literature.

Codd, E. (1970) A Relational Model for Large Shared Data Banks. *Communications of the Association for Computing Machinery*, v 13, n 6, p 377-387.

Informix (1994) *Illustra 2D Spatial DataBlade Guide (Release 1.3),* Illustra Information Technologies, Inc., Oakland CA, October 1994.

Informix (1994) *Illustra 3D Spatial DataBlade Guide (Release 1.2),* Illustra Information Technologies, Inc., Oakland CA, October 1994.

Milne, P., S. Milton, and J.L. Smith (1993) Geographical object-oriented databases --- a case study. *International Journal of Geographical Information Systems*, v. 7, n. 1, p. 39-55

Oracle (1996) *Oracle7 Spatial Data Option Reference and Administrator's Guide (Version 7.3.2),* Oracle Corporation, Redwood City, CA, April 1996.

ESRI (1995) *SDE The Spatial Data Engine User's Guide, version 2.0.* Environmental Systems Research Institute, Redlands, CA.

Strickland, T.M. (1994) Intersection of Relational and Object, *AM/FM International Proceedings.* p. 69-75.

van Oosterom, P.J.M. (1993) *Reactive Data Structures for Geographic Information Systems.* Oxford University Press, New York.

Viljlbrief, T. and P van Oosterom, The GEO++ System: an Extensible GIS, *Proceedings Fifth International Symposium on Spatial Data Handling*, p40-50, Charleston, South Carolina, August 1992.

Zhou, X. (1995) A Comparison of Oracle7 MultiDimension and ARC SDM for Point Data Retrieval.

A Systematic Strategy for High Performance GIS

Liujian Qian Donna J. Peuquet

Department of Geography
The Pennsylvania State University
University Park, PA 16802 USA
qian/peuquet@geog.psu.edu

ABSTRACT

High Performance Computing is becoming increasingly important to large GIS applications, where the ability to store and access huge amounts of social and environmental data is crucial. In this paper we propose a systematic strategy using what we term a *virtual grid*; a quadtree-based decomposition of space used for the balanced allocation of distributed storage space. We also introduce the concept of a *quadtree spatial signature* as a highly compact spatial index for storage and retrieval. Our proposal is generic in the sense that it addresses problems at all levels of a high performance spatial database system, from the physical implementation of data storage and access methods to the user level for spatial query and analysis, with the underlying parallel computing model being an increasingly popular Network of Workstations (NOW).

1 Introduction: Parallelism in Different Levels of GIS

Efficient handling of spatial data is a growing concern for GIS as the amount of data available, indeed necessary, for addressing urban and environmental issues continues to increase. Terrabytes of data are now available from various governmental agencies. All levels of a GIS, from data storage to in-memory spatial operations and algorithms, can benefit from efficient partitioning and parallel computing strategies. A number of articles have been published in the topic of parallel strategies for GIS. Most of these, however, have concentrated on individual spatial data structures or algorithms [Wag92], [DDA92]. The more broad-ranging problem toward High

Performance GIS (HPGIS), however, is how to partition large amounts of complex and often highly interrelated data so that multiple computers can each be assigned a fraction of the data and complete the task cooperatively and effectively regardless of the task.

The physical implementation of any file structure involves allocating disk storage in units of fixed size. These are called disk blocks, pages or buckets, depending upon the level of description. We will use the term buckets. Conceptually, a bucket is simply a unit of storage containing a number of data records. Significantly enhanced performance in accessing large amounts of data can be achieved if the multiple buckets needed to satisfy a single query can be accessed simultaneously (i.e., in parallel). This can be achieved by distributing the set of buckets representing an individual data layer over multiple physical storage units. For locationally-based queries, maximal efficiency is achieved when the data are evenly distributed among the buckets on the basis of their locational value. However, the geographic distribution of data elements is typically highly variable, and often very clustered. Moreover, geographic distributions tend to be variable over time. Most existing GISs do not presume any correspondence whatsoever between conceptual ordering and physical distribution in storage. Since most GISs store data in unordered pages of a file, some kind of spatial indexing must be employed in order to avoid the inspecting large portions of the database unnecessarily. The *virtual grid* organization described in this paper is proposed as an effective method for mapping of the geographical distribution of a data layer to a physical storage distribution. The method proposed has it's root in a balanced file structure called Grid File originally developed for a non-spatial context in [NH84].

Spatial indexing structures currently used for geographic databases include K-D-B trees, R-trees, Grid Files, and quadtrees, among many others (see [Sam90] for a thorough review). These indexing structures usually store key-pointer pairs where the 'key' is the identifying spatial attribute or shape, and 'pointer' is the address of the whole record stored in the unordered data file. The idea behind all spatial indexing schemes is to subdivide a large search space into multiple smaller search spaces, so that only those potentially relevant (and much smaller) parts need to be actually examined for given query predicates.

There are important differences among the ways various indexing structures split the search space. The two fundamental approaches, following the two basic types of geographic data models, can be described as space-based vs. object-based partitionings [NH84]. An index structure where partitions are designed to contain, and to not subdivide objects, such as in R-trees [Gut84], has object-based partitions. Other examples of this partitioning strategy are K-D trees [Ben75] where the partitioning boundaries are drawn based on the location of the point data being indexed.

If, on the other hand, the partition boundaries are drawn symmetric to all dimensions regardless of any particular object's location it is a space-based partitioning scheme. Quadtrees are the best-known example of this partitioning strategy. Another distinction can be made among space-based partitioning techniques depending on whether or not the partitioning is regular. The area quadtree is a regular space-based partition index, because it always splits an area into four equal parts. In contrast, the Grid File as adopted in another high-performance GIS context [CSZ93] is an irregular space-based partitioning scheme, since it divides the space into variable intervals along each dimension. Below is a figure that illustrates these different partitioning schemes. Figure 1(a) represents an example partitioning for an R-tree, where each rectangle is a minimum rectangle that contains a set of objects or smaller bounding rectangles. Figure 1(b) represents the partitioning for a K-D tree. The irregular spatial subdivision of a Grid File is depicted by Figure 1(c) while the Figure 1(d) illustrates the regular spatial subdivision of an area quadtree.

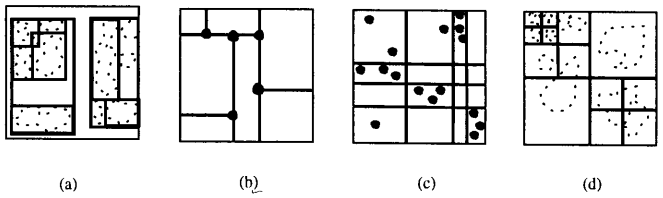


Figure 1: Patterns of Spatial Partitions

Although the R-tree, K-D tree and derivative methods have gained attention recently for indexing geographic databases for storage and retrieval, the area quadtree provides two significant advantages for parallel spatial indexing in a GIS context due to the regular subdivision of space. First, this allows for an even allocation of data records into buckets to be performed more easily. Although homogeneous areas or objects may be subdivided into different buckets for storage, the even allocation of data among multiple buckets becomes a more straightforward task. Second, the more direct mapping between geographic location and bucket allocation makes parallel retrieval of multiple layers on the basis of location a much simpler task. These features will be further explained in the discussion later.

2 Balanced Storage Allocation

In this section we describe the notion of the *virtual grid*, a quadtree based space decomposition and spatial data storage strategy.

2.1 Concept of the Virtual Grid Spatial Data Storage Model

The basic idea of the virtual grid is quite simple. The space or geographic area of interest is subdivided in the same manner as in normal area quadtree subdivisions. When to stop the decomposition is determined by the total volume of data within the resultant cells after each subdivision. Thus, an area with sparse data will result in few subdivisions and an area with a dense distribution will result in relatively more subdivisions. The result of the decomposition is a set of variably-sized tiles that covers entire area, with each tile corresponding to a leaf node in the quadtree. For each spatial tile, there is one data "bucket" associated with it that stores all the data which fall within the geographical area represented by that tile. We call our quadtree-based storage model a *virtual grid* for several reasons. First, the spatial resolutions of the stored data are not hierarchical, and the subdivision does not need to continue until only homogeneous spatial data are contained in the tile. The resulting set of partitions is a grid with variable-size tiles. Second, this irregular grid is a partitioning scheme only, with the individual tiles created by this partitioning potentially dispersed on different disks, as well as on different machines. Third, the subdivision bears no relation to the storage format used for the data themselves. The data within the subdivisions for any given data layer can be stored as vectors or pixels in accordance with the nature of the data.

Figure 2 illustrates the relationships between the individual tiles and data buckets for a given partitioning. The philosophy behind the use of the quadtree as a regular, hierarchical, location-based partitioning method stems from the simplicity of the area quadtree scheme, such as described in [Peu84]. The essential notion here is to provide a mechanism to even-out physical storage for what can be highly uneven and variable data distributions over geographic space, while still allowing efficient data retrieval for overlay and other layer-based operations.

Figure 2(a) represents a set of geographical data forming a layer of, say, land use on a set of islands. Figure 2(b) is the tiling of the corresponding layer, where there are 13 tiles numbered from 0 to 12. In figure 2 each tile is seen to correspond to a quadtree leaf node, with the whole quadtree (including both internal and leaf nodes) shown in Figure 2(c). If the data for a particular region are more dense than others, that region is always further divided into smaller tiles so that the volume of data contained within

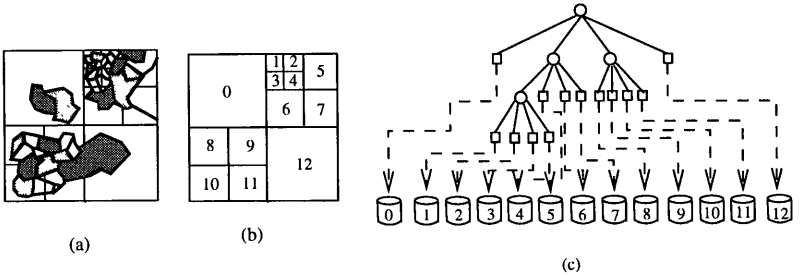


Figure 2: virtual grid, tiles, buckets and their relationships

any individual tile is approximately equal, regardless of the size of the geographical area represented.

2.2 Construction of a Virtual Grid

Now we consider how to construct a virtual grid for an individual layer of geographic data. It is worth mentioning that during the construction of the virtual grid we actually maintain a quadtree in memory to keep track of the levels and areas of the subdivision. As stated above, the level at which subdivision stops is totally determined by the amount of data contained within a tile. This amount is dictated by the size of a single bucket. All buckets are equal in their maximum size or capacity.

Given the bucket size, we now briefly describe how to store raw data records into a database in the sequence of Morton order, or equivalently speaking, how to the build the virtual grid (and hence the corresponding quadtree) in a bottom-up manner, in contrast to the normal top-down method.

The procedure is that we sort the raw data records first using the Morton order (or Z-order). This may involve calculating the Morton address for each record and sorting the records based on their addresses. Once sorted, we scan the records and add them sequentially into a bucket until it's capacity is reached. At this point we examine the Morton address of the last spatial object stored into the bucket, from which we are able to determine what is the smallest tile or quadtree node that should be associated with the bucket.

The following is an example showing how we load a set of spatial point data into a virtual grid using the bottom-up process.

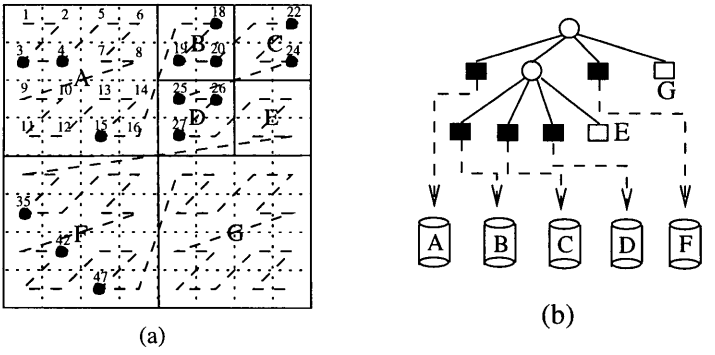


Figure 3: construction of a virtual grid

In Figure 3 we assume each bucket can hold 3 points, and the 15 data points to be stored have been labeled with Morton addresses and sorted. Upon the addition of the first three data points into a bucket, with the last point having a Morton address of 15, we form the corresponding tile A (which is associated with the bucket,) and it's three sibling quadrants, which have no buckets associated yet. Repeat this procedure and we eventually have 5 buckets with corresponding tiles being those black leaf nodes of the quadtree shown in figure 3.

The bucket size, as an important system parameter, has to be determined before raw data can be loaded into the database. As for most high performance database storage strategies, there is the issue of optimization: What bucket size provides the optimal performance given the distribution characteristics for a specific type of data? If the bucket size is too large, then the speed efficiency of doing operations in parallel is not fully realized. If the bucket size is too small, then the increased traffic of I/O operations caused by doing too many separate data access operations simultaneously can by itself slow down a computer, and indeed an entire computer network.

During the construction of a virtual grid, splitting of (large) objects may occur as a consequence of reallocating an overflow bucket. In our virtual grid storage model, when a bucket overflows, we subdivide it's corresponding tile into 4 equal quadrants; and reallocate the data in the bucket into at most four new buckets, depending on the data distribution in the original tile. During this subdivision and reallocation, if an object is large enough to cover more than one sub-tiles, we will split it and store partial objects into new buckets where they belong.

The last step of constructing a virtual grid will always be to construct a compact description of it, called a *quadtree spatial signature*, to be described in next section, and store the signature in the database catalog along with other meta information of a data layer.

3 Quadtree Spatial Signature

The whole idea behind the *virtual grid* notion is that when time-consuming operations over large volumes of data are to be performed in parallel, using a "divide and conquer" approach, we need to physically map buckets onto multiple physical storage areas in order to achieve three things; 1.) physically balance the load for large data access operations, 2.) provide an efficient indexing mechanism so that the physical location in storage of any given data element can be determined quickly and with a minimum amount of disk access, and 3.) when doing spatial join of multiple layers, we can have an intuitive approach to associate geographically-relevant buckets from different layers onto multiple computers, due to the nature of the *virtual grid.*

In this section we present our method for achieving this, utilizing something we call the *quadtree spatial signature.*

We define a *quadtree spatial signature* as a compact mapping of where data elements are located spatially into a search path within quadtree-space. Since quadtree subdivision was used to create the data tiles, this provides a quick index that quickly eliminates consideration of any geographical areas that are "blank" as far as the data in question. It also provides a rapid conversion from geographic space to storage location, and due to the regularity, it provides an intuitive method of assigning buckets of different data layers based on geographical relationships, which will greatly enhance the effectiveness of parallel processing of multiple layers.

Given a quadtree, it's spatial signature is simply a set of bitmap strings, with each level of the quadtree having one bitmap string, as shown in Figure 4. For any given level of a quadtree, the bitmap string is an array of value '00', '01' or '11's, based on the type of a corresponding tree node in that level. An internal node is labeled as '01', while empty leaf nodes are labeled as '00', and black leaf nodes (which have data in corresponding buckets) are '11's. Note that even if, for a particular level, there is no tree nodes at all, we still assign '00's to their corresponding positions in the bitmap array, because our bitmap strings represent a complete quadtree.

At a first glance the size of our signature may seem to be quite large, since it records information for every nodes of a complete quadtree. But in implementation one can always use compression methods such as run-length encoding to store the bitmap, as there are many repeated '0's or '1's; and the size of such a quadtree signature is actually quite small (about less than hundred KBytes for a quadtree of 10 levels).

Our spatial quadtree signature has the following properties:

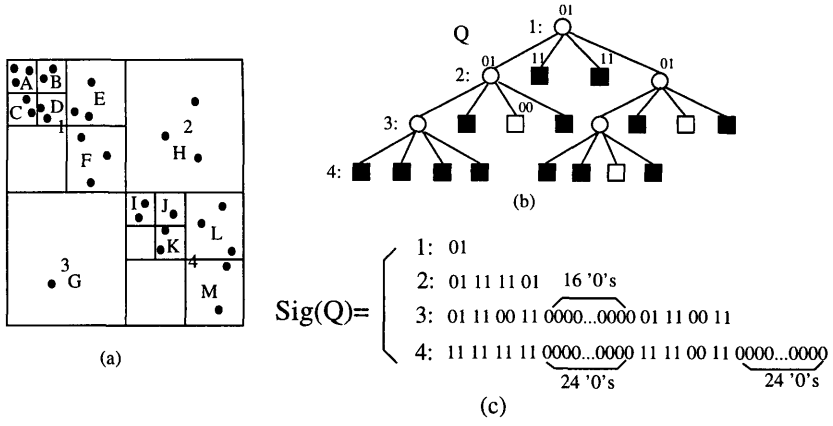


Figure 4: An example of quadtree spatial signature

- Morton addresses or Z-order numbers are used as an index inside the bitmap array of each level. The Morton address for each quadtree node in a given level is calculated using the spatial scale corresponding to that level. In this way the status of any tree node in that level can be obtained by calculating it's Morton address and index into the bitmap array. For example, in Figure 4, the bitmap string for level 2 is '01', '11', '11', '01', indicating that the four nodes (whose Morton addresses are labeled as 1,2,3,4 in Figure 4(a)) are internal nodes, leaf nodes, leaf nodes and internal nodes respectively.

- For an internal node in a higher level with Morton address N, the status of it's four quadrants can be obtained by looking up in the bitmap array of the next level at those positions from (N-1)*4 to N*4. Similar direct determination is possible from a child node to it's parent node.

- For a bitmap array, only those elements with value '11' are leaf nodes that have buckets associated with them. The number of tiles of a particular size in our virtual grid can be easily obtained by scanning the corresponding level's bitmap array and count the number of elements with value '11'.

- This set of bitmap arrays can serve as a spatial index; and mappings between '11'-valued elements (tiles) and actual storage buckets can be easily established based on the level number and array index for the elements. This removes the need for disk-based spatial indexing structures such as R-trees or K-D-B trees; and converts spatial search into in-memory calculations based on the set of bitmap arrays and searching rectangles (used in range search). The bitmap arrays also remove the need for maintaining expansive quadtree structures in

memory, since all the information has been retained in the set of bitmap arrays.

- The data distribution information at different spatial scales (tree levels) are maintained in the different levels of bitmap arrays. This is useful for scale-sensitive operations.

One of the most important rules to follow in applying parallel strategies to spatial operations, is that the load allocation algorithm must take into account not only how to balance the amount of data to be processed individually, but also how to assure that the data allocated to a computer are as geographically clustered as possible in data-space.

Below we will briefly describe how to use our quadtree spatial signature in balancing load that meets these two requirements, on a simple but increasingly popular parallel computing model of NOW (Network of off-the-shelf Workstations), where a master computer allocates data buckets to multiple slave computers so that spatial operations can be performed locally and concurrently by each slave on the protion of data it receives.

The main allocation procedure (for single-layer based operations), is that for each bitmap array, we maintain a cursor indicating the next un-allocated bucket (tile) in that level. Starting from the left-most '11' element of all the arrays, which is the first tile in our virtual grid, advance the cursor in that array for following consecutive '11's, until the limit of bucket number for one computer is reached or an element of different value is encountered. In the first case, we allocate the set of buckets traversed to the first slave, and continue for the next slave; in the second case, where we encounter a different type of element, we will need to look at the array either above or below the current array, based on whether the element with non-'11' value is an empty node or internal node. We then advance the cursor in the other array in a similar manner until we find the first '11' elements, and records all the consecutive elements of '11's as the set of buckets to be allocated to next available slave computer. We then repeat this step until all the buckets have been allocated; or the available slave computers are exhausted.

For the example virtual grid of Figure 4, if we assume each processor can handle at most 3 buckets of data, then using the above algorithm will result the sets of tiles like (A,B,C), (D,E,F), (G, H, I), (J,K,L), and (M). This allocation is well-balanced and presents a good level of geographical adjacency among tiles within each working set.

4 Conclusion

In this paper we presented a systematic strategy for high performance GISs that need to deal with very large data volumes. We proposed a *virtual grid* structure for distributing the storage of unevenly distributed spatial data in an even way; and suggested that a simple structure called the *quadtree spatial signature* can be effectively used in guiding the dividing of work load for time-consuming spatial operations. Finally, we want to emphasize that our method may also be extended into hexagonal and triangular tessellations as well.

References

[Ben75] J. L. Bentley. Multidimensional search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

[CSZ93] Mark Coyle, Shashi Shekhar, and Yvonne Zhou. Evaluation of disk allocation mehtods for parallelizing spatial queries on grid files. In *Data Engineering Conference*, 1993.

[DDA92] Yuemin Ding, Paul J. Densham, and Marc P. Armstrong. Parallel processing for network analysis: Decomposing shortest path algorithm for MIMD computers. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, pages 682–691, Charleston, South Carolina, 1992.

[Gut84] A. Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD Record*, 14, No. 2:47–57, 1984.

[NH84] J. Nievergelt and H. Hinterberger. The grid file: An adaptable, symmetric, multikey file strcuture. *ACM TODS*, 9(1):38–71, 1984.

[Peu84] Donna J. Peuquet. A conceptual framework and comparison of spatial data models. *Cartographica*, 21:66 –113, 1984.

[Sam90] Hanan Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, Mass., 1990.

[Wag92] Daniel F. Wagner. Synthetic test design for systematic evaluation of geographic information systems performance. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, pages 166–177, Charleston, South Carolina, 1992.

DEVELOPMENT OF A COMMON FRAMEWORK TO EXPRESS RASTER AND VECTOR DATASETS

J. Raul Ramirez, Ph.D.
Senior Research Scientist
The Ohio State University
Center for Mapping
Columbus, Ohio, USA

ABSTRACT

The most common ways to graphically represent geographically referenced data (geographic data) in computer-compatible form are raster and vector Conventionally, raster and vector are considered to be two different and independent ways to portray geographic space. Software programs are developed to deal only with raster or vector data. Of course, there are computer applications that allow the simultaneous display of raster and vector data, but internally, different software routines deal with each data type This is inefficient and costly. At The Ohio State University Center for Mapping, we have been working on the conceptualization of a highly advanced mapping system the Total Mapping System (TMS) One aspect of the TMS is data distribution In dealing with this topic, we decided to investigate some fundamental questions about data models. Are raster and vector really different types of data? If not, is there a common framework which expresses both datasets as unique data types? Is there a need for a different type of geographic representation (besides conventional raster and vector)? This paper presents the results of this research

BACKGROUND

There are many ongoing research efforts toward the development of new alternatives to conventional mapping The Total Mapping System (TMS) concept, in development at the Center for Mapping, is one of them. The TMS will support comprehensive real-time acquisition, processing, and distribution of up-to-date geographic information. The Airborne Integrated Mapping System (AIMS) is one component of the TMS and is currently in development at the Center for Mapping

The goal of the AIMS initiative is to develop a fully computer-compatible, real-time mapping system "capable of large-scale mapping and other precise positioning applications" (Bossler, 1996). This airborne system will integrate state-of-the-art positioning and imaging technology such as. differential GPS, INS, CCD, laser, and infrared sensors As indicated by Bossler (1996), the goals of AIMS are to. (1) acquire position and orientation of an aerial platform at 5-10 centimeters and ~10 arcsec, respectively, in real-time; (2) perform essential processing of digital images such as histogram equalization and imprinting in real-time, (3) generate dense ground control coverage in real-time, and (4) post-process digital imagery to calculate feature coordinates at submeter accuracy and to automatically recognize targets

The end product of AIMS will be ground images with a large number of three-dimensional ground control points generated in real-time. This will eliminate the current need for ground surveying and post-flight photogrammetric triangulation, and could provide very precise relief representation But, AIMS needs to be complemented with other research projects in order to achieve the goals of the TMS Of course, one major problem to be solved is the automatic extraction of terrain features from the remotely sensed images.

The major obstacle for the automatic extraction of features from remotely sensed images is the limited amount of explicit information in the images A possible solution to this problem is to increase the amount of explicit information per pixel This can be achieved by combining different sensors as part of a new data acquisition system such as AIMS. Additional sensors may be thermal cameras, laser profiler and imaging laser, SAR, SLAR, interferometric SAR, and/or multi- and hyper spectral scanners (Heipke and Toth, 1995) Using Figure 1 and Set notation. this concept could be expressed as follows

A conventional pixel carries today three pieces of information two planar coordinates (I,J) defining its location on the image, and an attribute The attribute is usuall, a graphic attribute, such as color This can be written as.

$$P = \{I, J, \text{Attribute}\} \quad (1)$$

Let us consider a different type of raster image,

$$R_N = \{P_{11}\ P_{12}\ P_{ij}...\}, \tag{2}$$

where P_{ij} indicates a particular pixel. Each pixel, besides the conventional information, has information generated from the different sensors For example,

$$P_{ij} = \{I, J, \text{Attribute}, \phi, \lambda, \text{elevation}, g_{ij}\ l_{ij}\ t_{ij}\ h_{ij} ...\}, \tag{3}$$

as defined in Figure 1 These ideas fundamentally change our conceptualization of raster data. Under this concept, pixels carry a rich amount of positional and attribute information It is expected that pixels belonging to the same terrain feature have a subset S (of P_{ij}) of common characteristics. With enough integrated sensors, it is possible that these characteristics are sufficient to automatically recognize the outline of each terrain feature

New mapping concepts such as the TMS of The Ohio State University Center for Mapping will radically change the field of geographic data generation. But, they will also change our ideas about data models It is clear that the current raster model will not be able to satisfy the needs of systems such as the TMS. Thus, we decided to study the problem of conventional geographic data models.

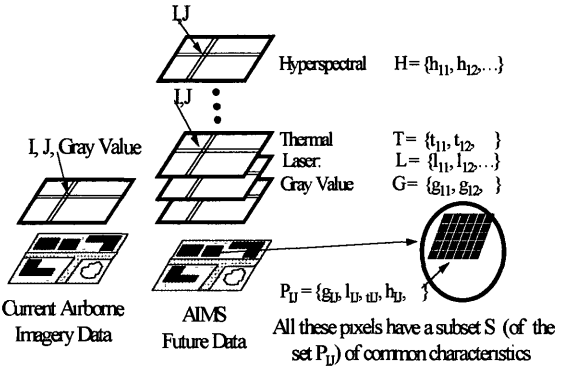


Figure 1. The TMS Acquisition Concept

THE ESSENCE OF RASTER AND VECTOR DATA MODELS

Let us define a two-dimensional geographic space to be represented in computer-compatible form Let us call this space **E**. Currently, there are two different models to express this space, the raster and the vector model The raster model is obtained by dividing the space **E** into a finite number of basic units. Each unit has a finite area and similar shape to all the others We will use the term **pixel** to designate the basic unit in the raster space The vector model is obtained by dividing **E** into an infinite number of area-less and dimension-less units We will call these basic units **geometric points.**

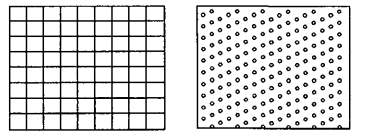


Figure 2 The Space **E** and its Basic Units

This concept can be extended to a three-dimensional space **V**, by dividing it in equal-size cubes (three-dimensional pixels) for the raster model, and three-dimensional geometric points for the vector model For simplicity the two dimensional model will be discussed here Figure 2 shows the space **E** and the basic units

Let us decrease the size of the pixels, as an example, by one-half of the original size In this case the number of pixels will increase from **n** to **4n** The result is shown in Figure 3.

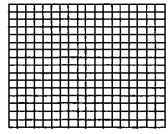


Figure 3 The Space **E** and a Smaller Raster Unit

If we repeat this process many times, in the limit, the number of pixels in the space **E** will be infinite and the raster and the vector model will be the same. Therefore,

$$\lim_{n \to \infty} (\text{Raster Model}) = \text{Vector Model} \qquad (4)$$

THE FUNDAMENTAL RELATIONSHIP

From the above result, we can argue that, in the limit, the raster and vector models are the same as shown by expression (4), and that raster and vector representation can be obtained from a single mathematical model. This global model is:

$$X = BU\,x, \quad Y = BU\,y, \qquad (5)$$

where

$$BU = \text{Distance}^{\,1}, \qquad (6)$$

BU = Basic Linear Unit

Distance = 1 (for vector model)
Distance > 1 (for raster model)

x = number of basic units in a primary direction (for example the X-axis)
y = number of basic units in the other primary direction (for example the Y-axis)

In the vector domain, BU is equal to one. This is equivalent to have a dimension-less area-less geometric point as the fundamental primitive. In the raster domain BU is greater than one (we assign a finite dimension to **Distance**). In this case the fundamental primitive is the pixel of length equal to **Distance** and area equal to **Distance**^{2}

As an example, Figure 4 shows the location of an arbitrary point **A,** for the case **Distance = 10** units (pixel length) Then, from formula (5)

BU = 10 x 1 = 10 (raster), BU = 1 x 1 = 1 (vector)

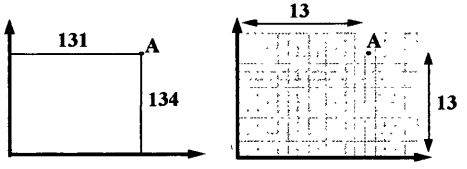


Figure 4 An Arbitrary Point In the Space **E.**

If, in expression (6), x = 13 and y = 13 for the raster representation and x = 131 and y = 134 for the vector representation of point **A**, then we have for the raster and vector spaces, respectively

$$X = 10 \quad 13 = 130, \quad Y = 10 \quad 13 = 130 \text{ (raster)}, \quad X = 1 \quad 131 = 131, \quad Y = 1 \quad 134 = 134 \text{ (vector)}$$

Expression (6) will reproduce the conventional expressions used in the raster and vector model by normalizing these equations by **BU**. In that case,

$$X = x, \quad Y = y,$$

and for Figure 4, we have.

$$X = 13, \quad Y = 13 \text{ (raster)}, \quad X = 131, \quad Y = 134 \text{ (vector)},$$

for the raster and vector cases, respectively

CARTESIAN DISTANCE, TRANSLATION, SCALING, AND ROTATION

The Cartesian distance between points **A** and **B**, shown in Figure 5, is given by,

$$d_{AB} = BU\,[\,(x_A - x_B)^2 + (y_A - y_B)^2\,]^{1/2} \tag{7}$$

A translation of the line **AB** is given by

$$\begin{aligned} X_A &= BU\,x_A + BU\ \ dx, \quad Y_A = BU\,y_A + BU\ \ dy \\ X_B &= BU\,x_B + BU\ \ dx, \quad Y_B = BU\,y_B + BU\ \ dy \end{aligned} \tag{8}$$

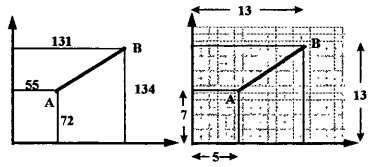


Figure 5. Distance **AB**

The scaling of the line **AB** by a factor **S** can be defined as follows,

$$d_S = S\,d_{AB}, \tag{9}$$

and the coordinate values of the end points of the scaled line are given by

$$\begin{aligned} X_A &= S\ BU\,x_A, \quad Y_A = S\ BU\,y_A \\ X_B &= S\,BU\,x_B, \quad Y_B = S\ BU\,y_B \end{aligned} \tag{10}$$

If we compute the coordinates of the line **AB** in a coordinate system rotated by an angle α, the resulting line is given by

$$\begin{aligned} X_{ar} &= BU\,x_A \cos\alpha + BU\,y_A \sin\alpha, \quad Y_{Ar} = BU\,y_A \cos\alpha - BU\,x_A \sin\alpha \\ X_{Br} &= BU\,x_B \cos\alpha + BU\,y_B \sin\alpha, \quad Y_{Br} = BU\,y_B \cos\alpha - BU\,x_B \sin\alpha \end{aligned} \tag{11}$$

DESCRIBING FEATURES IN THE GLOBAL MODEL

In the previous sections we presented a global model that encompasses the geometric aspect of the traditional raster and vector model The description provided by this model is equivalent to the skeletal representation developed by Ramirez (1991) for vector data. In order to provide a complete global model, three additional aspects need to be considered (1) a way to describe features in the raster model, (2) graphic variables (or graphic attributes) for raster and vector features, and (3) nongraphic attributes We will discuss them next

Description of Features in the Raster Model Traditionally, raster images are composed of "dumb" pixels "Dumb" pixels have no connectivity, or geometric or feature-related information Each pixel carries only positional two-dimensional (I,J) information and an attribute Orthophotos are a typical example of "dumb" pixel images. They show the surface of a particular area of the Earth in an orthogonal projection (distances and angles are equivalent to the one on the ground) They carry a large amount of implicit information but little explicit information Ideally, we would like to have images with "smart" pixels "Smart" pixels of an image carry a large amount of explicit information (similar in some fashion to the information in the vector model) The concept of "smart" pixels will be expanded in the following paragraphs

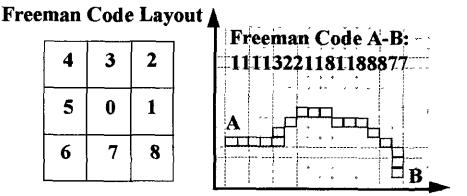


Figure 6. Freeman Code.

A simple way to define linear features in the raster model is using the Freeman code. The Freeman code carries connectivity and geometric information The skeletal representation of features in the raster model can be expressed by the Freeman code Figure 6 illustrates the description of skeletal representation of features using the Freeman code

In order to relate the Freeman code with expressions (5) through (11), let us assume the center of each pixel as the origin In that case, two different distances, as indicated in Figure 7, for BU (see expression (5)) need to be considered The distance between two pixels connected side by side is P. The distance between two pixels connected by a corner is equal to P $(2)^{½}$

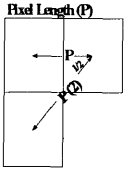


Figure 7 The Basic Distances in a 2-D Pixel

Therefore, expression (6) can be rewritten as·

$$BU = \text{Pixel Length, or } BU = \text{Pixel Length } (2)^{½}, \qquad (12)$$

and all the previous equations can be extended to express Freeman code relations For example, the Cartesian distance of the raster skeletal representation AB of Figure 6 is $d_s = 5P + 2(2)^{1/2}P + 2P + (2)^{1/2}P + 2P + 3(2)^{1/2}P + 2P = 19\,46\,P$ This distance is computed by subdividing the line into its seven straight segments, computing the length (in pixels) of each one, and adding them

Expanding the Freeman Code The traditional representations of features in raster images by Freeman codes, are still inadequate, by several factors of the equivalent vector representations Some of these factors (graphic variables or graphic attributes, and nongraphic attributes) were mentioned earlier. A factor not mentioned yet, is the geometric dimension of conventional raster data (two-dimensional) as opposite to spatial vector data (three-dimensional) A simple solution to this problem is to extend Freeman from two-dimensions to three This can be accomplished as follows (see Figure 8)

The planar representation of the Freeman code can be extended to a volumetric representation by considering the cube (instead of the square) as the fundamental representational unit. In this particular case, a three-dimensional pixel will have twenty-six (and only twenty-six) adjacent three-dimensional pixels These three-dimensional pixels are located at a level (Level 1) below the pixel of interest, at the same level (Level 2) as the pixel of interest, or at a level (Level 3) above the pixel of interest Figure 8 shows the basic unit (the three-dimensional pixel), the pixel of interest (pixel 10), the twenty-six adjacent pixels, and the three levels and identification number for each pixel

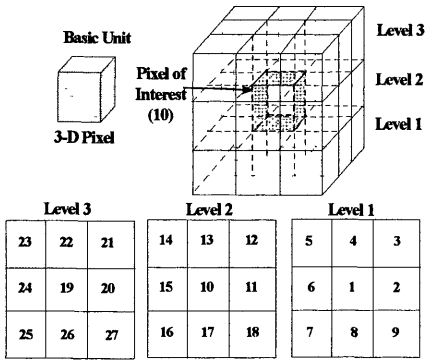


Figure 8. 3-D Freeman Code

Use of the Three-Dimensional Freeman Code Spatial features can be expressed by a three-dimensional Freeman code Figure 9 illustrates one such feature. The darker blocks indicate the skeletal representation of the feature A-B The three-dimensional Freeman code describing this feature is·

25 19 15 15 15 15 15 24 19 22 15 15 24 24

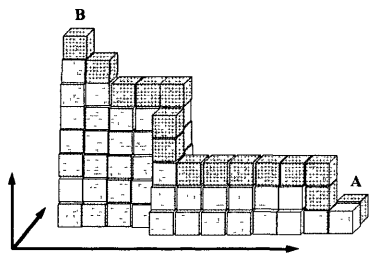


Figure 9. A 3-D Feature

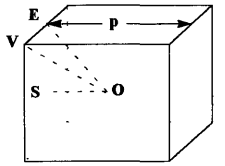
The length of this feature is computed by adding the space distances (Cartesian distances) of the different straight segments This is accomplished by expanding equation (3) to three dimensions, as follows·

$$d_{AB} = BU\,[\,(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2\,]^{1/2} \qquad (13)$$

The length of the feature AB of Figure 9 is d_{AB} = 16 37 P (applying expression (13)). This is the result of computing the length of the eight straight segments of AB (in function of P) and adding the results

For three-dimensional computations, an additional value for BU needs to be considered. Figure 10 illustrates this, where OV is the new value. Notice that OS, OE, and OV in this figure are one-half of the distances between the centers of the pixels

Graphic Attributes As discussed by Ramirez (1991), the graphic characteristics of a feature in the vector model are defined by Bertin's (1983) visual variables: space dimensions **(SD)**, size **(SI)**, value **(VA)**, patterning **(PA)**, color **(CO)**, orientation **(OR)**, and shape **(SH)** The space dimensions **(X, Y, Z)**, the spatial locations of any geometric outline (skeletal representation), are covered by the geometric discussion of the previous sections.



$OS = (1/2)\ p,\ OE = (1/2)\sqrt{2}\ p,$
$OV = (1/2)\sqrt{3}\ p$

Figure 10. The Basic Distances in the 3-D Pixel

The variable **size** uses the change in the dimensions of a graphic sign to communicate a specific meaning to it, for example, the width of a line as shown by Figure 11-a The variable **value** expresses the different degrees of grays used in the representation of a graphic sign Figure 11-b is an example The variable **patterning** represents the design or pattern used in the construction of a graphic sign. Line types, line symbols, cross-hatching, and area patterning are examples of this variable, its use is illustrated in Figure 11-c The variable **color** represents the use of colors in graphic signs to attach a specific meaning to them. For example, in a map, the color blue is used to indicate water Figure 11-d shows the outline of three buildings with the words red, blue, and green to indicate the color of each one The variable **orientation** uses the alignment of graphic point signs as a way of communicating a particular meaning to them, Figure 11-e illustrates this Finally, the variable **shape** uses the outline of a graphic point sign to represent a specific feature as demonstrated in Figure 11-f

In the raster model only the variables **value** or **color** are used to express the graphic characteristics of pixels In the case that the Freeman code is used to describe a feature, additional Bertin visual variables could be used, such as **size, patterning**, etc In general, we can state that Bertin's visual variables are enough to express the graphic attributes of features in the raster and vector model, including those cases where the Freeman code is used

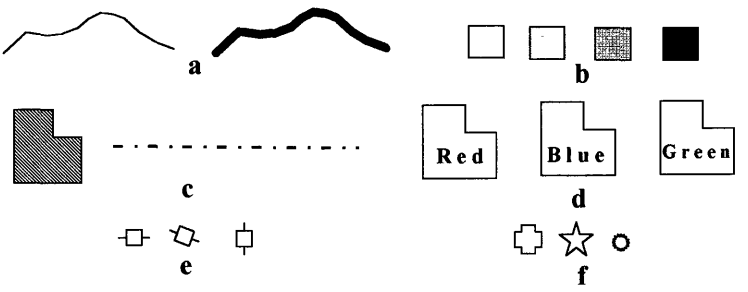


Figure 11 Bertin's Visual Variables

Nongraphic Attributes Nongraphic attributes are widely used in the vector model to carry additional information about the geographic features represented Typical examples of nongraphic attributes are the Digital Line Graph (DLG) codes (major and minors) used by the U S Geological Survey (USGS) Nongraphic attributes can be a combination of text and numeric characters In the case of the DLGs, for example, mostly numerical values are used The major code 50, for example, indicates that the corresponding feature belongs to the hydrographic category

In the raster model, in general, there are no nongraphic attributes at the individual pixel level In that case when the Freeman code is used to describe a feature, nongraphic attributes similar to the one for the vector model may exist

THE CURRENT STRUCTURE OF VECTOR AND RASTER MODELS

In the previous sections we have argued that raster and vector data can be expressed by a single global model and have proposed a framework that allows us to carry similar information for vector and raster ("smart" pixels) data On the other hand, we recognize that currently, raster and vector are considered different models and that software applications are developed for only one model In the following paragraphs we will present a summary of the major characteristics of practical implementations of the raster and vector data models today in order to understand better how the global model could be used. This will be followed by the outline of the new geographic data model

Raster and vector files carry positional and graphic information differently In the raster model, positional values are carried implicitly and graphic values (usually Bertin's **value** or **color** only) are carried explicitly for each **BU** From the viewpoint of the files' structure, generally in the raster model, there is one or more computer record (header) carrying common information for the geographic area represented In these records, at least the size of the **BU** and the extent of the area are defined Then, a value is carried for each **BU** (ignoring in this discussion any attempt to compress the data) Data are stored by rows or columns (not by feature).

In the vector model, positional values are carried explicitly by **significant points** Significant points are those positional points needed to define a feature uniquely, for example, the end points of a straight line, the center point and two arc points (and a direction convention) of an arc. Positional values may describe a complete feature, or different segments of a feature, depending on the data organization (spaghetti vs topology). Graphic values are carried per feature (not per **BU**) and, generally, there is a header with common information.

For the vector model, nongraphic attributes are combined with the positional, and the graphic attributes, in some cases, or in other cases may be combined only with the graphic attributes, or they may be stored in a different file.

It is obvious that current raster files are unable to carry the wealth of information of images of the future. On the other hand, vector files that have a less rigorous structure may be able to carry all type of additional information But, this will always require the conversion of raster information into vector. Either situation is not ideal. We want to be able to use all the information of the images of the future directly This is our motivation to present next the outline of a new geographic data model

THE OUTLINE OF A NEW GEOGRAPHIC DATA MODEL

It was indicated earlier that new mapping concepts are in development One of them, the TMS, at the Ohio State University Center for Mapping, will support real-time acquisition of raster images. It was also pointed out that such a concept will integrate many different sensors which provide additional information per pixel. It is expected that combining these pieces of information will allow the user to develop specific signatures for the identification of terrain features. All of this will be done in a highly automated fashion

Because digital images will be acquired, and because each pixel of these images will have a large amount of new information (compared with current pixels), it makes sense to consider an alternative to conventional raster and vector data models The new model must be raster based but it should have "smart" pixels "Smart" pixels are three-dimensional primitives which will allow the user to perform, in raster images, similar manipulations to those performed with current raster images, plus those manipulation and queries performed with vector data. Ideally, this model should allow the generation of images with "dumb" pixels only, "smart" pixels only, or with a combination of "dumb" and "smart" pixels.

The fundamental relationships for this model will be given by expressions (5) and (6), extended to a three-dimensional space. The raster primitives will be point, line, area, and volume. Their skeletal representations will be expressed by three-dimensional Freeman codes (see Figure 12 for examples of some of these primitives) Graphic and nongraphic attributes will be attached to them In this framework, these primitives will be used to generate more complex elements. Terrain features will be described as a whole, or as a set of segments. The last description will allow topological structuring of features

In this model we can conceptualize each pixel as a cube carrying information about the surrounding pixels and about the terrain it represents. Information will be spatial information in the form of spatial coordinates in a user-selected system,

connectivity to other pixels, topologic relationships, graphic attributes, and nongraphic attributes.

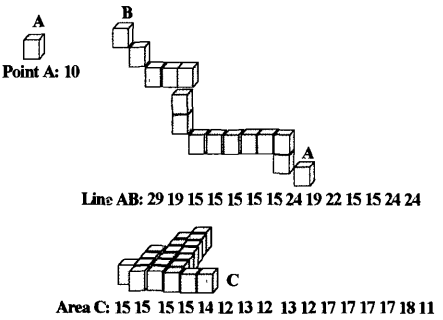


Figure 12 Point, Line, and Area Basic Raster Primitives

CONCLUSIONS

A conceptual framework to express raster and vector datasets has been presented This framework has been called a "global" model and allows us to express locations and geometric relation in both raster and vector domain by a single set of expressions From this, a framework for a new raster data model with "smart" three-dimensional pixels has been proposed. This raster model allows us to perform all current raster and vector manipulations and queries This new format will satisfy the requirement of the mapping systems of the future

The ideas presented here are being implemented at the Center for Mapping The new raster model is implemented as an extension of the Center for Mapping Database Form (Ramirez, at al., 1991), (Bidoshi, 1995)

REFERENCES

Bidoshi, K. (1995) Application of the Center for Mapping's Database Form (CFMDBF) for Evaluating DLG Road Coverages Using GPSVan Data, Internal Report, The Ohio State University Center for Mapping

Bossler, J D. (1996) Airborne Integrated Mapping System. An Initiative of the Ohio State University, *GIM*, Vol 10, 32-25

Heipke, Ch., Toth, Ch. (1995). Draft for Design of Total Mapping System, Internal Document, The Ohio State University Center for Mapping

Ramirez, J R (1991) Development of a Cartographic Language, *Proceedings COSIT'91.*

Ramirez, J.R., Fernandez-Falcon, E Schmidley. R , Szakas, J (1991) The Center for Mapping Database Form, Internal Report, The Ohio State University Center for Mapping

MEDIAL AXIS GENERALISATION OF HYDROLOGY NETWORKS

Michael McAllister Jack Snoeyink
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4

Abstract

We examine some benefits of using the medial axis as a centreline for rivers and lakes. One obvious benefit, automatic centreline generation, has been used for many years. We look at how the topological relationships between the medial axis and the river banks or lake shores can provide extra network characteristics such as river areas and opposite river banks. We also report on our experience at approximating the medial axis with a Voronoi diagram of point sites.

1 Introduction

Maps are rich in geometric structure. Adjacencies between features, containment in regions, intersections of lines or regions, and relative orientations or proximities of features all contribute to the value of maps and map operations. For digital maps, much of this structure is lost to computers; the visual cues that people use to see the geometric structure are not available to a computer. Instead, we develop algorithms for topology building, polygon containment, and polygon intersection to capture this structure for map analysis by computers.

Structure, in particular locality and proximity, appears in various forms. Regular grids [21] and quad trees [12, 19] localise points into a small region of space. The medial axis describes the "shape" of polygons in a variety of fields such as map labelling [2], shape matching [13, 15], solid modelling [24], mesh generation [9], and pocket machining [10]. Voronoi diagrams [3, 8] capture both the locality of objects as well as their proximity to one another for applications such as identifying polygon closures and line intersections while digitising from maps [7]. In this paper, we focus on the medial axis.

Centrelines have been a standard tool of manual cartography for generalising networks for many years. Digital cartography inherits centrelines for generalisation of river and road systems [14], for simplifying the analysis of these same systems, and for extracting linear features from raster models [17]. The medial axis

is one method that practitioners have used to generate these centrelines automatically.

The characteristics that make the medial axis a good choice for centrelines can be taken one step further for river networks. An edge of a river's the medial axis is the bisector of the two nearest river banks. Therefore, each medial edge identifies a pair of river banks that are nearest to one another. This nearness relationship allows us to

- associate opposite banks of a river.
- tie analysis on centreline networks to original river bank data.
- calculate surface areas for rivers and river segments.
- extend network orderings, such as the Strahler [22] or the Horton [11] orders on river networks, to include lakes and wide rivers for cartographic generalisation.

Although the medial axis is a well-defined structure, calculating the structure in the presence of degeneracies can be difficult. Consequently, we use an approximation to the medial axis in our experiments. The approximation is based on a robust implementation of the Voronoi diagram for points.

In section 2 we describe our motivation for looking at centrelines of river networks. Section 3 provides some basic geometric definitions. Section 4 gives a few more details on the benefits of the medial axis as a centreline. Finally, our approximation to the medial axis by a Voronoi diagram of points appears in section 5.

The work in this paper has been done in conjunction with Facet Decision Systems.

2 The Problem

River slope, shore length, and surface area are three characteristics that influence the suitability of a river for salmon spawning. In digital maps, rivers appear as a single-line or as a set of river banks. In the first case, river slope and shore length come directly from the single-line rivers and a surface area estimate comes from some nominal width for the river. In the second case, the river is defined implicitly by its banks. We can compute slope and length for individual river banks, but there is no correlation between opposite banks. As with single-line rivers, a nominal width for the river generates an approximation to the surface area, but the approach ignores the implicit information of the map, namely the delineation of the river itself.

River centrelines lead to a better estimate of both length and area for wide rivers. Centrelines are a common approach to converting hydrology networks into tree-like river networks [14]. The length of the centreline averages-out the difference in length of the two river banks as the river meanders. Moreover, a centreline with flow-directed edges establishes upstream and downstream relationships between tributaries on opposite sides of the rivers.

The area of a wide river is a bit more elusive than the length. Although river banks may be labelled as either a right or left bank, digital maps do not usually encode which portion of a river bank is opposite another bank. Consequently, we know where the river is, to the left or right of an edge, but we don't know how wide the river is. The key to getting a better area estimate beyond using a nominal river width lies with finding the centrelines automatically and with making better use of the rivers' centrelines.

In our system, river centrelines are a subset of the medial axis of the river polygons. Efficient algorithms [1, 6] can compute the medial axis of a river automatically and generate river slopes and lengths. As a by-product of the algorithm, each edge of the medial axis is tagged with the two closest river banks and each river banks is tagged with its nearest medial edge. Consequently, given a point on a river bank, we can find the distance from this point to its nearest medial axis edge; this distance is half of the river's width at that point. Knowing the width of the river at any point of our choosing allows us to make a better estimate of a river's surface area.

3 Definitions

The *medial axis* of a polygon [4, 18] consists of the centre of all the circles contained inside the polygon that touch two or more different polygon edges. Polygon vertices, where two edges meet, are counted as a single edge. For any such circle, its centre is equidistant to the two edges that it touches and is therefore on the *bisector* of the two edges (figure 1).

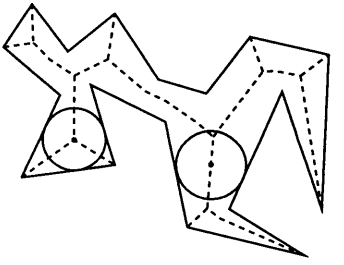


Figure 1: Medial axis of a polygon (dotted) and two touching circles.

The Voronoi diagram [3, 16] of a set of point *sites* is a partition of the plane into maximally-connected regions in which all points share the same nearest sites. Points that are equidistant to two nearest sites form the boundaries of the partition's cells. The cell boundaries, called *Voronoi edges*, are bisectors between the closest sites. Points that are equidistant to three or more nearest sites are called *Voronoi vertices*. In non-degenerate cases, the Voronoi vertices are defined by three sites; if you join the three sites that define a Voronoi vertex then you obtain a *Delaunay triangle*. The same definition of a Voronoi diagram holds when line segments, such as the edges of a polygon, or arcs are the sites instead of points.

4 Medial Axis Benefits

The key aspect of the medial axis in Section 2 is the association between the edges of the medial axis and the nearest banks of the river. This association provides

more benefits than simply an estimate of river widths.

First, the association of the medial axis edges to their nearest river banks tie future calculations on the centreline to original data elements. The river centrelines replace the river banks in a network to give a single-line river network. Further analysis, such as identifying drainage basins, locating fish spawning habitat, and tracking the run-off of forest cut blocks, uses the single-line network. Attributes of the medial axis edges from these operations are propagated to the appropriate river banks. The single-line network allows for simpler network analysis without sacrificing links to the river banks.

Second, the medial axis edges define opposite banks. Since a medial axis edge is the bisector of its closest river banks, these two banks are, in effect, opposite one another along the river. Attributes of opposite banks such as slope, elevation, soil type, and vegetation type can then be compared, to detect either inconsistencies of the data or anomalies in the environment.

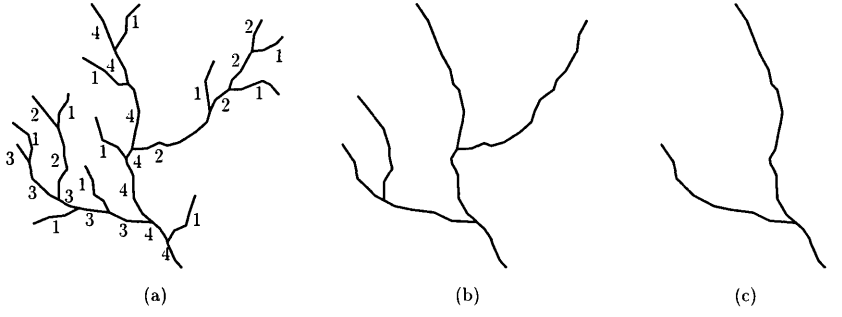


Figure 2: A river network with its Horton order (a) and the subnetworks with Horton numbers greater than 1 (b) and 2 (c).

Third, the association extends network orderings to wide rivers and lakes for cartographic generalisation. The medial axis or centrelines of rivers have long been used to generalise rivers [14]. Network orders on single-line networks, such as the Horton [11], Strahler [22], and Shreve [20] orders, extract the primary branches of a river network for generalisation at large map scales. Figure 2 shows a sample network with its Horton order and the result of selecting edges of only high order from the network. These same orderings can be applied to networks that contain lakes or river banks by treating the lakes and wide rivers as their medial axis; this is not surprising. The original lake shore edges receive an order number from the nearby medial edges. Selecting edges with high network orders will also extract the lakes along the path. A minimum area for the lake is of added benefit at large map scales.

The propagation of network orders to lake shores and river banks can take one of two forms. Lake shores receive the network order of the nearest medial edge or lake shores receive the network order of the highest medial edge in the lake. The latter form treats a lake as a single unit to preserve the visual cues of lake extent

and shape and is our preference for network simplifications. (Not all edges of a centreline have the same network order number.)

5 Computation of the Medial Axis

The computation of the medial axis is well-studied in computational geometry. Optimal algorithms [1, 6] have been published to compute the structure for simple polygons. The medial axis is also a subset of the Voronoi diagram of the polygon's edge and, as such, algorithms that compute Voronoi diagrams [5, 16, 18, 25] are applicable to finding the medial axis.

Unfortunately, few implementations for computing the medial axis of a polygon available are robust. Although many Voronoi diagram algorithm implementations exist, most do not handle the "degeneracies" of lines that share common endpoints, which arise when you try to compute the medial axis as a subset of a Voronoi diagram. Consequently, we use a robust sweep algorithm for the Voronoi diagram of point sites to approximate the medial axis of a polygon.

5.1 Medial Axis Approximation

Given the polygonal contour of a river or lake, we discretise the boundary of the river, compute the Voronoi diagram of these points, and approximate the medial axis from the result. Theoretically, as more points are added to discretise the boundary of the polygon, the Voronoi diagram inside the polygon converges to a superset of the polygon's medial axis. Computationally, adding more points to the boundary adds degeneracies and increases the computation time. We need to strike a balance between computation time and diagram fidelity.

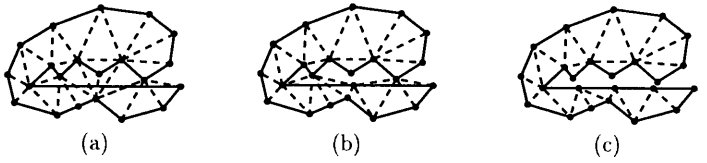


Figure 3: Delaunay triangles inside the polygon after (a) the initial point, (b) one decomposition step, and (c) two decomposition steps.

Our solution adaptively discretises the river boundary and starts with the points in the river bank's polygonal lines (figure 3(a)). After computing the Voronoi diagram of these points, we compare each Delaunay triangle of the Voronoi diagram with the boundary of the river: if some river boundary cuts through any Delaunay triangle then we split the edge at its midpoint, add the midpoint to the set of point sites and recompute the Voronoi diagram (figure 3(b)). The result of these iterations is a decomposition of the river's interior into Delaunay triangles (figures 3(c) and 4).

We do not want the entire medial axis as the centreline of a river. We only want those edges that lead to tributaries or that link tributaries; we consider the outflow end of a river to be a tributary. Consequently, we mark all of the Delaunay triangles that have a tributary at one of its vertices and mark the Delaunay triangles inside the river that form a path between the tributaries' triangles.

There are two ways to approximate the medial axis from the marked triangulation. The first uses the subset of the Voronoi diagram whose vertices correspond to marked Delaunay triangles. If the initial discretisation of the river edges resulted in a good triangulation and the points along an edge were far apart from one another then this approximation can look like a zig-zag pattern rather than an expected smooth centreline.

The second uses a representative point inside each Delaunay triangle and joins the points of adjacent marked triangles into paths. Of course, the result of this method depends on the choice of representative points. One possibility uses the centroid of each triangle. If the base of the triangles alternates between river banks and the triangles have one side much smaller than the other two sides, then the approximation is jagged. Another possibility uses the midpoint of the line between the middle of the shortest triangle edge and its opposite triangle vertex as a representative point and produces a much smoother effect for long and thin triangles. In both cases, the approximation has a tree structure and both the Voronoi edges and the Delaunay triangles record the closest river bank edges.

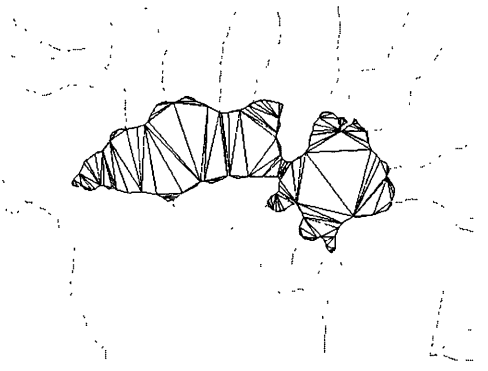


Figure 4: Delaunay triangles inside of a lake.

The medial axis itself does not address the entire problem; the direction of water flow along the medial axis edges has been ignored so far. A correct direction of flow is important when you want to answer queries such as "What is the river area upstream from a particular point?" or "If a particular tributary is polluted, what (downstream) fish spawning habitats may be affected?" In most cases for rivers, the direction of flow for the medial axis edges matches either the flow direction of a tributary at one end of the edge or the flow direction on the river banks that define the medial edge. This does not apply to medial axis edges inside lakes since the lake shore edges do not contain any flow. For lakes, a few simple topological rules have been sufficient in our experiments:

- if an edge is adjacent to a tributary then the flow of the edge matches the flow of the tributary.
- if the medial axis edges inside a lake meet at a node then there must always be at least one edge that enters and at least one edge that leaves the node.

We set the direction of flow on medial edges that lead directly to tributaries according to the first rule before examining the inner edges of the medial axis.

5.2 Area Generation

As mentioned in Section 2, the area of a river can be derived from the river's length and width. When Voronoi edges approximate the medial axis, width measurements are the distance between Voronoi edges and river banks. When paths between Delaunay triangles approximate the medial axis, we obtain the area in a different manner: assign the area of each Delaunay triangle to its representative point. Since the Delaunay triangulation decomposes the interior of the river, the river area between two points is the sum of the areas at the approximation's nodes between the two points.

Computing river area from Delaunay triangles has some drawbacks. Not every Delaunay triangle has its representative point in the approximate medial axis since the approximation only keeps the portions of the medial axis that link tributaries. As seen in figure 5, the areas of some inlets and bays must be allocated to a nearby representative points to preserve all of the feature's area. The variation in granularity of the triangle areas is another drawback. The area of a triangle in a river branch may be small while the area of a triangle at a river junction may be large.

6 Sample Centrelines

Our data is supplied by the Canadian Department of Fisheries and Oceans and Facet Decision Systems. It is a set of coded polygonal lines that outline terrain features. We use the hydrological features: rivers, river banks, and lake shores. The data is grouped in 1:20 000 scale map sheets with a 1 metre accuracy in the xy-plane and a 5 metre accuracy in elevation. The data adheres to the 1:20 000 TRIM data standard of British Columbia [23]: rivers and river banks are digitised in a downstream direction while lake shores are digitised in a clockwise direction. Rivers whose width is less than 20 metres are digitised as the centreline of the river. Rivers whose width exceeds 20 metres have their left and right banks digitised; no association between opposite banks appears in the source data. Although the polygonal lines are not guaranteed to appear in any particular order, the digitising standard mandates two characteristics: polygonal lines only meet at their endpoints, which are numerically identical.

Since the polygonal lines are unordered, we must build the topology of the data before computing the medial axis of the features. Adjacent lines share numerically identical endpoints so we place all the line endpoints into two-dimensional buckets and then use matching points within each bucket to find adjoining edges. The matched ends provide enough topology to trace the outline of lakes and rivers.

We have extracted the directed centrelines and areas of features in the mountainous interior of British Columbia where lakes have few out-flowing rivers. In

Figure 5: The area of inlets and bays must be assigned to nearby medial edges or vertices.

the 500 lakes and rivers tested, the resulting water flow has been consistent with the expected flow in all of the cases. In the majority of the cases, the lakes only had one outlet and one inlet so deriving the direction of flow is simple. Other rivers or lakes, as in figure 6, have a medial axis that branches more than off just one centreline where the outlets were grouped at one end of the lake. The grouping makes the general water flow patterns simpler and more predictable than a lake with widely distributed outlets.

Although we obtained area estimates for the lakes and rivers in the tested watersheds, the process was not without difficulties. While the data digitising standard seemed ideal for geometric computations, we still needed to find and remove inconsistencies— primarily digitising errors: open polygons, miscoded edges, reversed edges, and missing edges.

Another difficulty, which we have not yet resolved, is the over-estimate of the area caused by islands and sandbars. Sandbars appear along river banks and narrow the effective width of the river. Islands eliminate area from the river. We expect to handle sandbars by using a more liberal definition of a river bank. As for islands, we can subtract their area from the rivers or lakes to which they belong, but this solution is not very satisfactory; it does not give us an easy method for finding the area of a river between two points on the river banks, and the automatically-generated centrelines do not necessarily respect the land formations (figure 6).

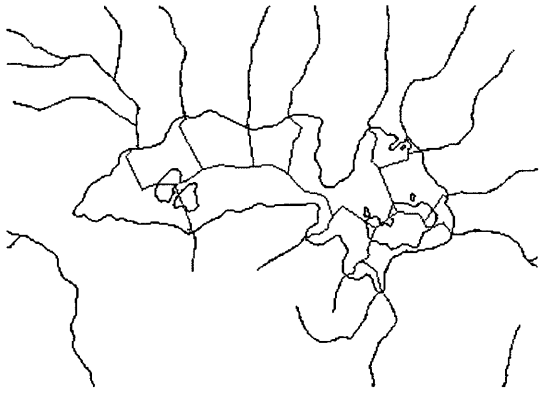


Figure 6: The medial axis of the boundary does not respect islands.

Acknowledgements

The authors thank Facet Decision Systems for their support of this work, both financial and technical. We also thank the Canadian Department of Fisheries and Oceans for the use of their hydrological data and the British Columbia Advanced Systems Institute for their financial support.

References

[1] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete & Computational Geometry*, 4:591–604, 1989.

[2] J. Ahn and H. Freeman. A program for automatic name placement. In *AutoCarto 6*, pages 444–453, 1983.

[3] F. Aurenhammer. Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[4] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[5] J.-D. Boissonnat, O. Devillers, and M. Teillaud. An semidynamic construction of higher-order Voronoi diagrams and its randomized analysis. *Algorithmica*, 9:329–356, 1993.

[6] F. Chin, J. Snoeyink, and C.-A. Wang. Finding the medial axis of a simple polygon in linear time. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput. (ISAAC 95)*, volume 1004 of *Lecture Notes in Computer Science*, pages 382–391. Springer-Verlag, 1995.

[7] C. M. Gold. Persistent spatial relations: a systems design objective. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 219–225, 1994.

[8] C. M. Gold. Three approaches to automated topology, and how computational geometry helps. In *Proceedings of the 6th International Symposium on Spatial Data Handling*, pages 145–156. IGU Commision on GIS, 1994.

[9] H. N. Gürsoy and N. M. Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part I algorithm. *Engineering with Computers*, 8:121–137, 1992.

[10] M. Held. *On the Computational Geometry of Pocket Machining*. Number 500 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[11] R. E. Horton. Erosional development of streams and their drainage basins—hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin*, 56:275–370, 1945.

[12] T. J. Ibbs and A. Stevens. Quadtree storage of vector data. *International Journal of Geographical Information Systems*, 2(1):43–56, 1988.

[13] N. Mayya and V. T. Rajan. An efficient shape representation scheme using Voronoi skeletons. *Pattern Recognition Letters*, 16:147–160, 1995.

[14] B. G. Nickerson. Automated cartographic generalization for linear features. *Cartographica*, 25(3):15–66, 1988.

[15] R. L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

[16] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, 1992.

[17] D. Peuquet. An examination of techniques for reformatting digital cartographic data/part 1: The raster to vector process. *Cartographica*, 18(1):34–48, 1981.

[18] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, 1985.

[19] C. A. Shaffer, H. Samet, and R. C. Nelson. QUILT: a geographic information system based on quadtrees. *International Journal of Geographical Information Systems*, 4(2):103–131, 1990.

[20] R. L. Shreve. Statistical law of stream numbers. *Journal of Geology*, 74:17–37, 1966.

[21] A. K. Skidmore. Terrain position as mapped from a gridded digital elevation model. *International Journal Of Geographical Information Systems*, 4(1):33–49, 1990.

[22] A. N. Strahler. Quantitative analysis of watershed geomorphology. *Transactions of the American Geophysical Union*, 8(6):913–920, 1957.

[23] Surveys and Resource Mapping Branch. *British Columbia Specifications and Guidelines for Geomatics. Digital Baseline Mapping at 1:20 000*, volume 3. Ministry of Environment, Lands, and Parks, Province of British Columbia, Jan. 1992. Release 2.0.

[24] P. Vermeer. Two-dimensional MAT to boundary conversion. In *Proc. 2nd Symp. Solid Model. Appl.*, pages 493–494, 1993.

[25] C. K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete & Computational Geometry*, 2:365–393, 1987.

VISUALIZING CARTOGRAPHIC GENERALIZATION

Robert B. McMaster and Howard Veregin
Department of Geography
414 Social Science Building
University of Minnesota
Minneapolis, MN 55455

ABSTRACT

Map generalization is a complicated and not well-understood process. In order for users to go beyond the simple and arbitrary application of generalization operators, and accompanying tolerance values, to features in databases, visualization techniques must be designed to allow a better understanding of the effects. Through such visualization users can quickly understand, and adjust, their decisions. This paper proposes, given an existing framework of the generalization process in both vector and raster mode, a series of techniques for visualizing the outcome. Several graphic examples of such techniques are provided.

INTRODUCTION

As evidenced by the growing literature and number of conferences on the topic, interest in automated map generalization is rapidly growing. The reason for this increased attention to map generalization is obvious: with well-structured spatial databases now commonly available (such as TIGER), the creation of multiple representations from these master databases, and more complex types of spatial analyses, require sophisticated capability to generalize these data. Complete automated generalization remains one of the major unsolved problems in digital cartography and GIS.

Multiple operators for the generalization of digital databases have been developed and fully tested in both a vector and raster format (Rieger and Coulson, 1993). The model detailed by McMaster and Shea (1992) establishes a classification of both vector-based (point, line, and area), and raster-based (structural, numerical, numerical categorization, and categorical) operations. Vector-based generalization includes operations such as simplification, smoothing, enhancement, and displacement; sample raster operations include gap bridge, erode smooth, and aggregation. Unfortunately, for both types of

operations, methods currently do not exist for visualizing the results of the generalization process. For instance, simplification involves filtering unnecessary coordinate data from the line; yet the user, other than through rather subjective visual comparisons and static geometric measures (McMaster, 1986), does not have analytical or visual methods for ascertaining the effect, or quality, of the simplification. In this paper methods are presented for visualizing the effect of several generalization processes. For instance, a common generalization practice would be the simultaneous application of simplification, smoothing, and displacement operations to features (McMaster, 1989). However, different algorithms and parameters might be applied to each feature with current interactive software. To visualize these processes, one approach might alter the *hue* of each feature with the operator, *saturation* with number of iterations, and *value* with tolerance level. Features that are both significantly simplified (=blue) and moderately smoothed (=red) would appear as a mixture of the colors, but with a higher blue content. Multiple iterations of an operation (i.e. smoothing) would increase (red) saturation. Other possible visual variables used to view generalization include size (parameter level) and texture (operation).

VISUALIZING VECTOR-MODE GENERALIZATION

Several frameworks for the organization of vector-based generalization operations have been presented. For this paper we will use the framework presented by McMaster and Shea (1992), which identifies the fundamental operations of *simplification, smoothing, aggregation, amalgamation, merge, collapse, refinement, typification, exaggeration, enhancement, and displacement.* To graphically understand the effects of a generalization process, a set of basic visual techniques must be established. For this purpose, we turn to the work of Bertin and the well-established visual variables. However, not all visual variables are appropriate, and addition visual methods must be designed. This may be seen more clearly with the operations of merge and simplification.

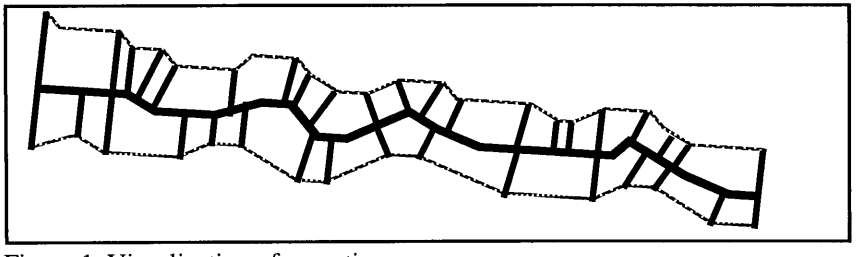


Figure 1. Visualization of operation merge.

Merge and Simplification

For the generalization operation **merge,** one simple approach for visualization involves creating a "millipede" graphic (Figure 1), where the legs depict distance to the original two boundaries that have been fused together.

Figure 2 depicts the potential use of value in visualizing the effect of a **simplification** operation. Simplification is a basic data reduction technique, and is often confused with the broader process of generalization. Simplification algorithms do not modify, transform, or manipulate x-y coordinates, they simpy eliminate those coordinates not considered critical for retaining the characteristic shape of a feature. A visual technique, then, must impart the idea of information loss.

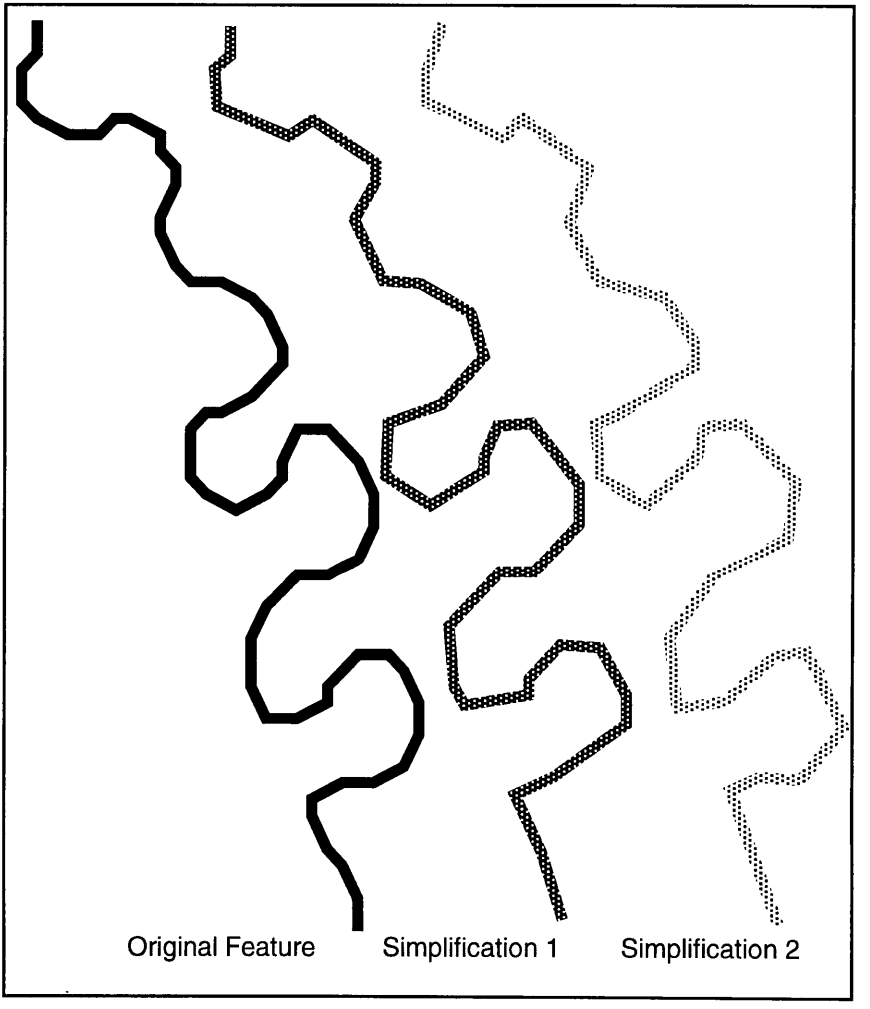


Figure 2. Use of value to depict increasing simplification

Here such visual variables such as size, value and saturation seem appropriate. This illustration uses value, where, as coordinates are eliminated through the

process of simplification, the line begins to "fade", representing fewer coordinates along the line. Other visual techniques might include depicting the displacement vectors, or areas lost, through the application of a simplification operation. Such a technique is illustrated on Figure 3.

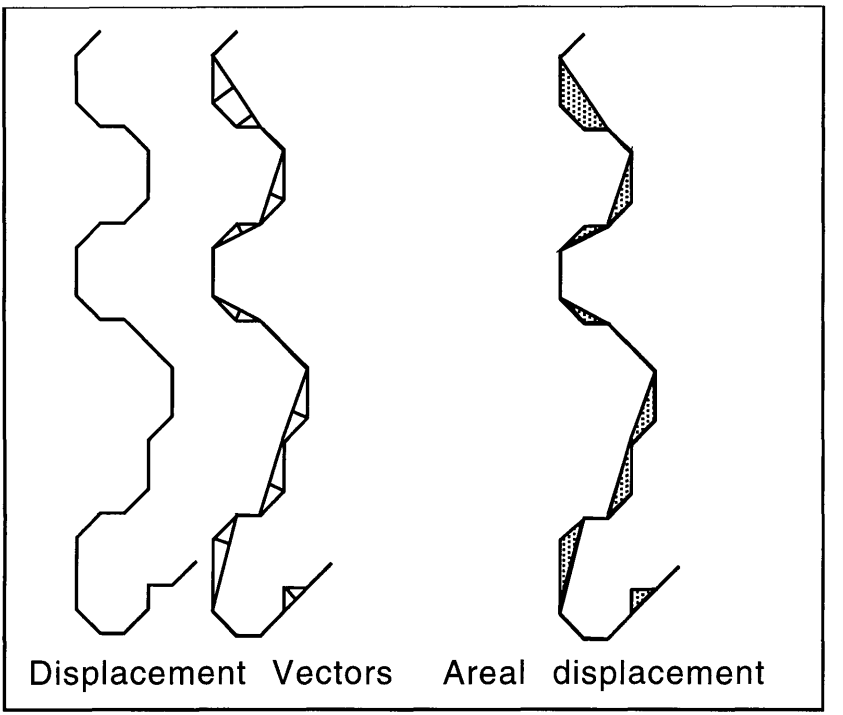


Figure 3. Visualization of displacement vectors and areal displacement for the operation of simplification.

Amalgamation

A series of visual techniques for depicting the generalization operation of **amalgamation** are also possible. Amalgamation involves the fusing together of polygonal features--such as a series of lakes, islands, or even closely-related forest stands--due to scale reduction. By fusing the features together, the intervening feature space is lost. It is this change in the ratio of feature **a** to feature **b**, as well as a sense of the original configuration of features, that is of interest to the user. One potential technique (Figure 4) involves the creation of a spider diagram that connects the multiple centroids of the original set of polygonal features and the centroid of the newly generated amalgamated polygon, thus illustrating the general complexity of the process. Another technique involves the application of value, where those regions of the lost category are emphasized, giving a sense of area change. Similar techniques could be applied to the process of **aggregation**, where a series of point features are fused into an

areal feature, normally represented by an enclosing boundary.

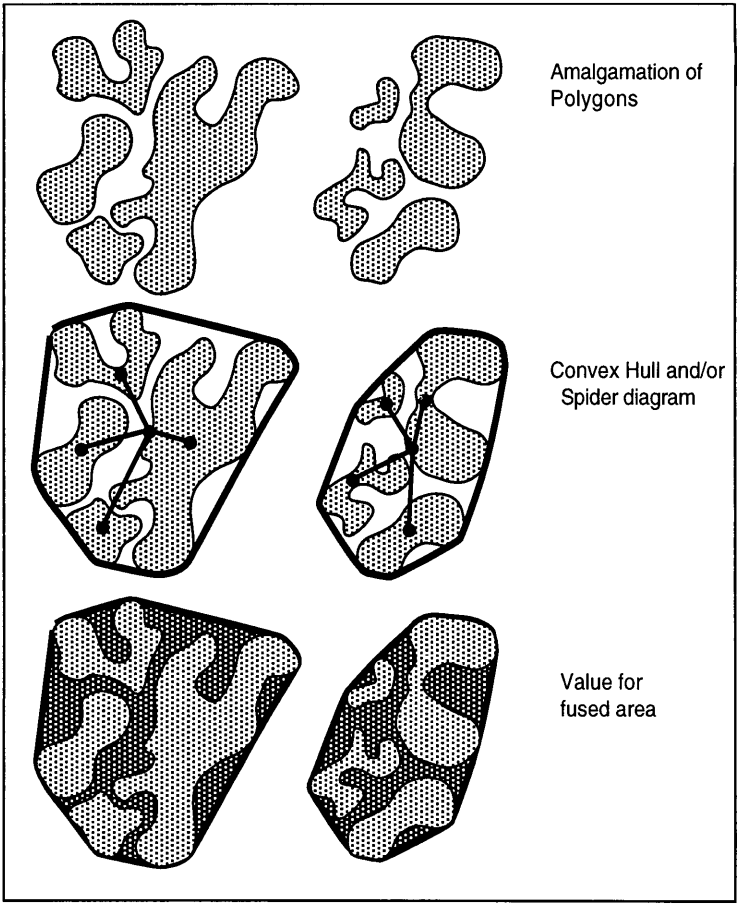


Figure 4. Visualization techniques for amalgamation

Smoothing

Possible techiques for the generalization operation of **smoothing** involve display of both displacement, as with simplification using displacement vectors and area, and changes in the angularity and curvilinearity of the feature. In this case changes in the value (Figure 5) of a linear feature represent decreasing angularity or curvilinearity. Another possible visual technique is to connect the inflection points of the curve in order to give a sense of how rapidly the curve is changing. Many short segments would indicate a more complicated line.

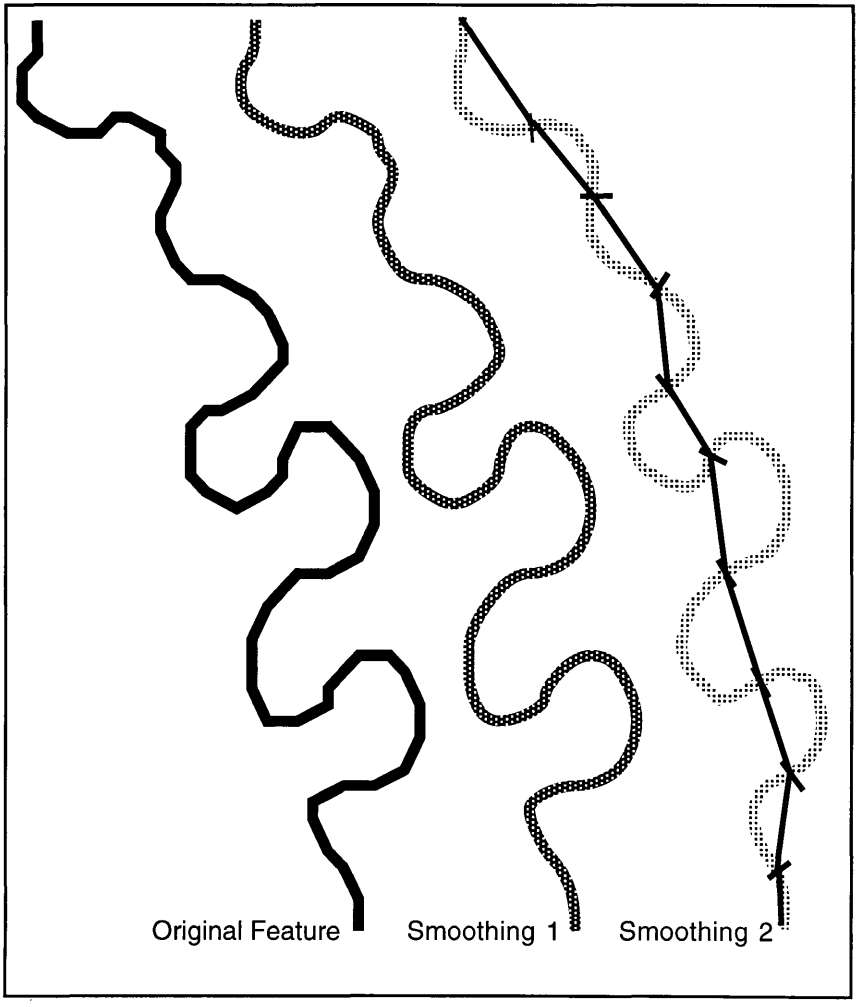


Figure 5. Techniques for visualizing angularity /curvilinearity changes resulting from smoothing.

VISUALIZING RASTER-MODE GENERALIZATION

Raster-Mode Generalization

A model developed by McMaster and Monmonier (1989) identified four basic classes of raster-mode generalization operators. The four fundamental categories developed in this framework included: (1) structural, (2) numerical, (3) numerical categorization, and (4) categorical generalization. Schylberg (1993) modified this framework to include a category of object-based raster generalization. A summary of these methods is provided in the original paper,

and are outlined below.

The first category, structural generalization, redefines the framework of the raster matrix, normally through a transformation of the cell-to-area relationship. The second, numerical raster generalization, also known as spatial filtering, manipulates the complexity of the image by smoothing deviations, or reducing the variance of the matrix, enhancing edges, or sharpening edges, as well as a variety of specific operations, such as those used to extract specific terrain classes. The third, numerical categorization, involves the classification of digital images using standard techniques from digital image processing in order to produce a classified 'categorical' output matrix. In the last category, purely categorical generalization techniques must reduce the spatial complexity through the consideration of the number or weighting of the attribute categories within the moving kernel. Such methods are intrinsically more difficult since the assignment of new values, based on nominal-level information, is a more subjective process.

As with vector-mode operations, a series of techniques for visualizing the effect of raster-mode generalization have been developed. For example, one may superimpose a set of original grid lines on a structurally modified image to display resolution change. For categorical generalization, one may use a saturation mask in order to show the effects of aggregation or erode smoothing, where the cells that have been modified are more saturated and those unchanged are less.

Other visual techniques can be applied to visualize the effects of generalization in raster mode. For instance, one common method used to generalize raster images involves the aggregation of cells into larger units (Figure 6). In this example each two x two matrix of cells is aggregated into one larger cell. The three **A**s and 1 **B** in the upper left portion of the matrix are aggregated to an **A** in the generalized version. However, some of the new cells are "stronger" than others in that the dominance of the original attributes varies. In this example, the three **A**s and one **B** represent a three-cell dominance. In the upper right area of the image, a four-cell dominance of the original category **D** is found. The integer dominance value can be represented with a value, texture, or saturation in a third image design to visualize the strength in any cell. The method here involves the use of value. The darker values represent those cells where, in the original image, greater numbers of the dominant cell were found. Such a technique would allow the user to differentiate those regions on the image where large blocks of cells were relatively homogenious versus those regions with a high spatial variance.

Additionally, by using only one image color could be effectively used to display the original categories, with saturation or value applied to represent the actual dominance value.

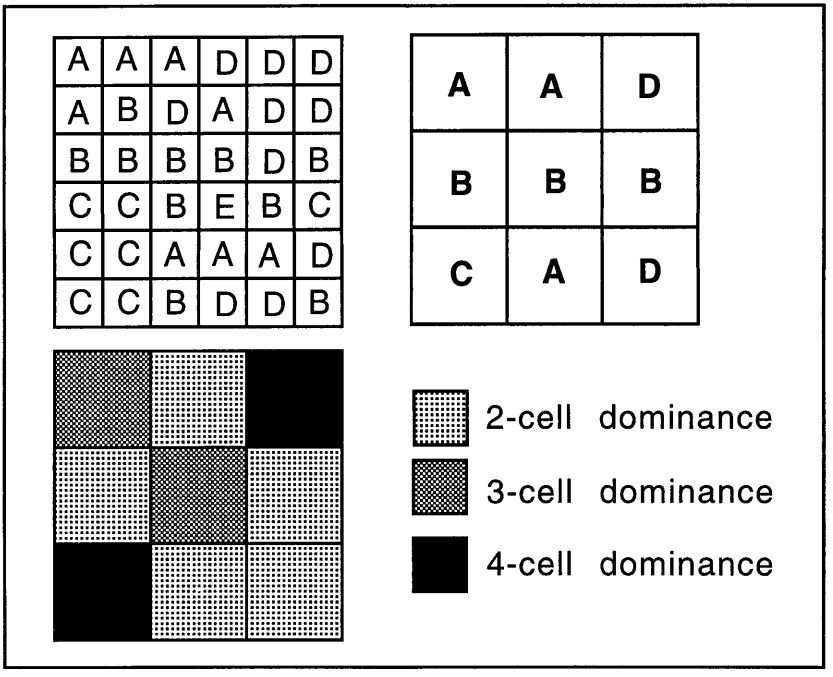


Figure 6. Method for depicting quality of cell aggregation.

Another visual technique for the raster-mode process involves the computation of a change matrix. In Figure 7 the right-hand land use / land cover matrix has been generated from the left-hand matrix using a simple modal filter. As a result of the application of the modal filter, a series of changes are made to the cells. The lower-left matrix numerically depicts these results where, for instance, in three instances category **A** was changed to category **D**. As with the previous example, these numerical results can be converted to a visualization where value is used to illustrate lower and higher changes.

A series of techniques may also be designed for *numerical* raster generalization, such as image processing and terrain modeling. The result of any generalization process on these data, such as low- and high-pass filtering, will be a numerical difference between the old and new image. To visualize change, a three-dimensional surface might be created and, due to the potential complexity, smoothed. Of course a third matrix, using an appropriate visual technique, could also be created, as described above.

With the use of color, actual numerical differences between the new and old cells could be displayed with value and the variance of the differences around a cell could be represented with saturation. For terrain models, this would allow for the display of changes in both elevation (value) and slope simultaneously.

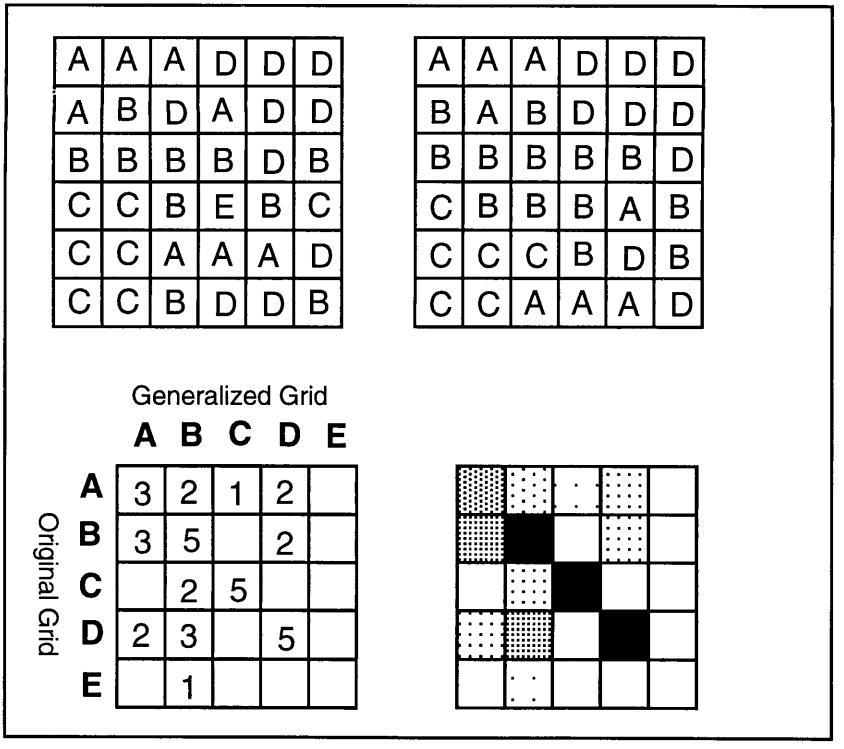


Figure 7. Method for visualizing category change on raster images.

SUMMARY

Research into automated generalization is now at a stage where, given recent progress, we can look beyond the actual operations and begin to focus on the resulting feature changes--the quality of the generalization. This will involve the design and implementation of appropriate visualization techniques, the integration of these techniques into existing interfaces for generalization, and user testing. Although we propose a preliminary set of methods in this paper, a more thorough set of techniques must be developed to cover all aspects of the process. Our work includes this further development as well as the creation of a conceptual framework for visualizing generalization based on existing frameworks of the process.

Perhaps one of the more important questions involves user-interface design. Several papers have reported on user interface design for generalization (McMaster, 1995; McMaster and Chang, 1993; and McMaster and Mark, 1991), but none have addressed the need for integrating visualization techniques. After generalization, the user must be provided through the interface with multiple options for viewing, and thus analyzing, the effect of the process. Given that many interfaces for generalization are in the design stage, it is timely to consider

the integration of such visualization methods before such interface design is finalized.

REFERENCES

Buttenfield, B.P., and McMaster, R.B. (Eds) (1991). *Map Generalization: Making Rules for Knowledge Representation.* Longman, London.

McMaster, Robert B. (1995). Knowledge acquisition for cartographic generalization: experimental method. In Muller, J.C, Lagrange, J-P, and Weibel, R, (Eds) *GIS and Generalization: Methodology and Practice.* Taylor and Francis, London, pp. 161 - 179.

McMaster, Robert B. and Chang, H. (1993). Interface design and knowledge acquisition for cartographic generalization. In: *Proceedings, Eleventh International Symposium on Computer-Assisted Cartography*, Minneapolis, MN, Oct, 1993, pp. 187-96.

McMaster, Robert B. and K. Stuart Shea. (1992). *Generalization in Digital Cartography.* Association of American Geographers, Washington.

McMaster, Robert B. and Mark, D. M. (1991). The design of a graphical user interface for knowledge acquisition in cartographic generalization. In: *Proceedings GIS/LIS'91*, Atlanta, GA, pp. 311-20.

McMaster, Robert B. (1989). The Integration of Simplification and Smoothing Routines in Line Generalization. *Cartographica*, 26(1): 101-121.

McMaster, Robert B. and Mark Monmonier. (1988). A Conceptual Framework for Quantitative and Qualitative Raster-Mode Generalization. In: *Proceedings*, GIS/LIS'89, Orlando, Florida. Falls Church, VA: American Society for Photogrammetry and Remote Sensing.

McMaster, Robert B. (1986). A Statistical Analysis of Mathematical Measures for Linear Simplification. *The American Cartographer,* 13(2): 330-346.

Rieger, Monika and Michael R. C. Coulson. (1993). Consensus or Confusion: Cartographers' Knowledge of Generalization. *Cartographica*, 30(2 & 3): 69-80.

Schylberg, Lars. (1993). Computational methods for generalization of cartographic data in a raster environment, Doctoral thesis, Royal Institute of Technology, Department of Geodesy and Photogrammetry, Stockholm, Sweden.

CARTOGRAPHIC GUIDELINES ON THE VISUALIZATION OF ATTRIBUTE ACCURACY

Michael Leitner
Department of Geography and Anthropology
Louisiana State University, Baton Rouge, LA 70803
email mleitne@unix1.sncc.lsu.edu

Barbara P. Buttenfield
Department of Geography
University of Colorado, Boulder, CO 80309
email babs@colorado.edu

This research establishes cartographic guidelines for mapping one specific aspect of data quality, namely attribute accuracy. The guidelines were derived through an empirical study in which test subjects simulated two different siting decisions on a CRT: the location of a natural conservation park and the location of an airport. Both siting tasks needed to incorporate the spatial distribution of wetlands that were dominant in the selected study area. The main goal of the experiment was to find out, if the correctness, the timing, and the confidence of both siting decisions varied due to the inclusion of certainty information about the wetlands locations, the number of classes for wetlands, and the graphical treatment of certainty information about the wetlands locations.

This research continues the trend re-establishing empirical testing as a valid paradigm for eliciting and formalizing cartographic design knowledge. The symbolization schemes for attribute accuracy developed in this paper should be incorporated as GIS graphical defaults in anticipation of digital datasets that include data quality information. Such cartographic guidelines, if expressed in the form of production rules, can also be implemented in expert systems for cartographic design. Such expert systems currently lack formalized knowledge about cartographic design principles, especially in the realm of symbolization.

INTRODUCTION

A major impediment to the development of a full-scale expert system in cartographic design is the lack of formalized knowledge about cartographic design principles (Buttenfield and Mark, 1991). Formalizing cartographic design principles requires acquisition and re-expression in the form of semantic nets, frames, production rules, or similar formalization methods. Techniques for the acquisition of cartographic knowledge for automated map generalization have been summarized by Weibel (1995). They include conventional knowledge engineering through interviews and observation of practitioners on the job or on artificial problems and reverse engineering, which tries to recapitulate decisions made on published documents or maps. Unfortunately, the techniques for cartographic knowledge acquisition have been discussed almost exclusively on a theoretical level and to date, little empirical research has

been conducted. One of the few empirical studies, applying reverse engineering, was conducted by Leitner and Buttenfield (1995). They utilized a computer-assisted inventory of the Austrian National Topographic Map Series in order to reconstruct the decisions made by cartographers during map compilation. The cartographic knowledge acquired in this study revealed quantitative relations between map elements (e.g., settlement, transportation, and hydrography) and the changes in these relations that occur with scale transition.

The lack of formalized cartographic knowledge is due to the fact that formalization usually involves empirical research (e.g., interviews, text analysis, inventory, etc.) that is in most cases very tedious and time-consuming. Complicating this matter is the fact that some design variables, such as visual balance and contrast, are by their very nature difficult to formalize. Textbooks (e.g., Robinson et al., 1984; Dent, 1996) usually offer some guidelines for such aesthetic issues, but these guidelines have not been expressed numerically nor have they been stated in form of rules. Another fertile yet untapped area for expert systems lies in the realm of symbolization. Expert systems have been successfully developed for map production, especially for displacement of map features and label placement of feature labels.

This paper reports on the elicitation of guidelines for mapping one specific aspect of data quality, namely attribute accuracy. The guidelines are based on empirical testing. This is the first time (to the knowledge of the authors), that guidelines about the use of attribute accuracy in maps have been based on an empirical investigation. Attribute accuracy as defined in the US Federal Information Processing Standard (FIPS) 173 refers to discrepancies in categorization or the probabilities of misclassification. Together with positional accuracy, logical consistency, completeness and lineage, attribute accuracy constitutes data quality (FGDC, 1992). In the context of this research attribute accuracy is parameterized in the following way: An attribute is confirmed at a given location by demonstrating that the same location is attributed identically in a second independent data source of equal or better quality. For example, one may be more certain of a location whose land cover is attributed to 'wetlands' by two databases, than of a location attributed to 'wetlands' in one database, but not in the other. Throughout this paper, attribute accuracy and attribute certainty will be used interchangeably.

Guidelines for the visualization of data quality and of attribute accuracy in specifically have been mainly discussed on a theoretical level. In many instances, the starting point in such discussions is Bertin's six graphic variables (Bertin, 1983) and how these variables (with possible additions or modifications) might be logically matched with different components of data quality (Buttenfield 1991, MacEachren 1992, van der Wel et al. 1994). MacEachren (1992) for example states that Bertin's variables size and color value are most appropriate for depicting uncertainty in numerical information, while color hue, shape, and perhaps orientation can be used for uncertainty in nominal information. Although not included in Bertin's original variables, color saturation and 'focus' can also be used for depicting uncertainty. Saturation can be varied from pure hues for very certain information to unsaturated (i.e., gray)

hues for uncertain information. 'Focus' refers to the visual sharpness of a symbol. Presenting data 'out of focus', or at lower spatial resolution, might be an ideal way to depict uncertainty (MacEachren, 1992).

Van der Wel et al. (1994) present a framework for the visualization of quality information in which they correlate graphic variables with data quality components at different levels of measurement. According to this framework, the graphic variables associated with attribute accuracy are color hue, size, texture, value and color saturation. A similar framework was proposed by Buttenfield (1991), in which the relationship between the quality components and data types is established through differing graphic variables.

The above approaches for displaying data quality encode quality information in an implicit manner (McGranaghan, 1993). In general, implicit symbology for data quality uses graphic ambiguity to create visual and cognitive ambiguity related to uncertainty in the data. Attribute ambiguity, for example, could be encoded by symbols which blend or mask the character of attributes. Fuzzy or indistinct symbols and animation are further approaches to create graphical ambiguity (McGranaghan, 1993).

It can be anticipated that merging data with their quality information into compound symbols (or what McGranaghan (1993) refers to as 'implicit symbology') make maps more complex and difficult to read. This issue is complicated by the fact that people are not used to having data quality information incorporated in a map display. On the other hand, the addition of interactivity, animation, and sound (Fischer 1994a, 1994b) opens up several possibilities for providing attribute information without interfering with the visibility of features that are present in the display (MacEachren, 1992). The results of this research however indicate that attribute accuracy, if applied appropriately can be embedded in maps without confusing map readers. It would seem that map certainty information is understood as clarification rather than adding complexity to a map display.

EXERIMENTAL DESIGN

The cartographic guidelines derived in this paper stem from an empirical study that investigated the impact of attribute accuracy displays of wetland areas on spatial decision support. During an experiment test subjects were asked to site a park and subsequently an airport in an under-developed region. The experiment was designed to observe how both decisions were made, in terms of how often decisions were made correctly, how long the decisions took, and how confident the subjects were about their siting choices. Multiple trials of the experiment varied the inclusion/exclusion of certainty about wetlands locations, the number of classes for wetlands (one or three), and the graphical treatment of certainty (by value, saturation, or texture). In each trial, one of eight possible test maps were randomly presented to each test subject.

The eight test maps differ in the depiction of wetland areas. The scale, study area, and base map information remain the same. One test map (hereafter

Map1) shows a single wetland class. Map1 can be characterized by low attribute detail and no certainty information about wetland locations. Another test map (hereafter Map3) depicts three different wetland classes (fresh water marsh, lake, and lagoon). Compared with Map1, Map3 possesses more detail, but no certainty information. The other six test maps display attribute certainty for wetland locations in two classes (more certain and less certain). One pair symbolizes certainty by varying texture (MapT and MapTi), a second pair varies value (MapV and Map Vi) and a third pair varies saturation (MapS and MapSi). MapT, MapV and MapS depict more certainty by (respectively) darker value, finer texture, and more saturated color. The subscript i indicates reversed symbolization, with more certainty shown by (respectively) lighter value, coarser texture, and more pastel color. Pairs of value and saturation were determined by pre-tests. Texture pairs were drawn from Zirbel (1978).

The experiment was set up in MacroMediaDirector running on Power Macintoshes 7100/80. Sixty-eight test subjects participated in the experiment and their responses were collected on-the-fly. A detailed description of the experiment, including the test maps can be found in Leitner and Buttenfield (1996) and Leitner (1997).

SYMBOLIZATION SCHEMES FOR MOST CORRECT, FASTEST AND MOST CONFIDENT SITING DECISIONS

The performance of many GIS applications for spatial decision support is dependent on the correctness of the decision, the speed with which the decision is made, and the confidence level after having made the decision.

The following discussion intends to establish empirical evidence documenting graphical guidelines that may be incorporated as GIS system defaults for mapping attribute accuracy. The first section discusses which visual variable(s) shall be selected when the correctness of the siting decision is of foremost importance. The following section suggests visualization guidelines yielding the fastest siting decisions. The last section discusses symbolization schemes that should be applied in order to achieve most confident siting decisions. Since space is limited, only the results with respect to the park location will be discussed in this paper. The results for the selection of the airport location can be found in Leitner (1997).

Symbolization schemes for making correct siting decisions

The Friedman-Test was calculated between Map1 and Map3; between Map1 and each of the six certainty maps (in pairs); between all six certainty maps (in pairs); and between all six certainty maps at the same time. The results of the Friedman-Test are shown in Table 1. When comparing pairs of certainty maps, only the statistically significant results are shown in Table 1. The first value in the 'mean rank' column refers to the first test map in the 'test maps' column, the second value, to the second test map. A higher frequency of correct siting decisions are indicated by a lower 'mean rank'.

TEST MAPS	MEAN RANK	CHI-SQUARE	D.F.	SIGN.
Map_1 and Map_3	1.54 / 1.46	0.8182	1	0.3657
Map_1 and Map_S	1.54 / 1.46	1.2857	1	0.2568
Map_1 and Map_{Si}	1.54 / 1.46	0.8182	1	0.3657
Map_1 and Map_V	1.47 / 1.53	0.4000	1	0.5271
Map_1 and Map_{Vi}	1.60 / 1.40	5.4444	1	**0.0196**
Map_1 and Map_T	1.57 / 1.43	2.7778	1	**0.0956**
Map_1 and Map_{Ti}	1.53 / 1.47	0.3333	1	0.5637
Map_V and Map_{Vi}	1.63 / 1.37	7.3636	1	**0.0067**
Map_S and Map_V	1.43 / 1.57	7.3636	1	**0.0588**
Map_{Si} and Map_{Vi}	1.54 / 1.46	3.0000	1	**0.0833**
Map_V and Map_T	1.60 / 1.40	3.7692	1	**0.0522**
Map_{Vi} and Map_{Ti}	1.43 / 1.57	3.5714	1	**0.0588**
All Certainty Maps	*	12.5893	5	**0.0275**

The table entries show the results of the Friedman-Test. When two certainty maps are compared, only statistically significant results are shown.

D.F. : degrees of freedom; **SIGN.:** level of significance

* The mean ranks are: Map_{Vi}=3.16; Map_T=3.32; Map_{Si}=3.46; Map_S=3.50; Map_{Ti}=3.60; Map_V=3.96

Table 1:

Correctness of Park Selections Compared for Different Test Maps

Overall, the results show that an increase in attribute detail translates into more correct answers for the park selection. This is true whether the increase in attribute detail relates to an increase in more attribute classes or to the addition of certainty information. However, only MapVi yields statistically significant improvement over Map1 at the 0.95 confidence interval. This is also true for MapT, but at a lower confidence interval (0.9). The results of the Friedman-Test for the category 'all certainty maps' (last line in Table 1) indicate statistically significant differences for the number of correct siting choices between the six certainty maps. The lowest mean rank among the certainty maps is yielded by MapVi (3.16), the second lowest by MapT (3.32).

If more classes of attribute data are available, than they should be displayed in the map. If attribute certainty information is available than it should be included in the map. If the choices of depicting certainty information are by saturation, value, or texture, than value should be selected. More certain information should be visualized by lighter value. When value is not available, texture should be applied, such as that more certain information should be depicted with finer texture and less certain information with coarser texture.

Source of Variation	Total Variation	D.F.	F	SIGN.
Map_1 versus Map_3	1053	1	4.638	**0.035**
Sequence	11011	3	16.167	0.000
Map_1 versus Map_S	390	1	1.483	0.228
Sequence	9044	3	11.475	0.000
Map_1 versus Map_{Si}	159	1	1.114	0.295
Sequence	8701	3	16.585	0.000
Map_1 versus Map_V	57	1	0.193	0.662
Sequence	5476	3	6.186	0.001
Map_1 versus Map_{Vi}	19	1	0.092	0.763
Sequence	7472	3	11.907	0.000
Map_1 versus Map_T	2	1	0.008	0.929
Sequence	10314	3	18.052	0.000
Map_1 versus Map_{Ti}	295	1	0.982	0.326
Sequence	10254	3	11.386	0.000
Map_S versus Map_{Si}	1135	1	5.012	**0.029**
Sequence	7523	3	11.069	0.000
Map_{Si} versus Map_{Ti}	969	1	3.669	**0.060**
Sequence	8824	3	11.140	0.000
All Certainty Maps	2650	5	1.239	0.293
Sequence	18805	3	23.526	0.000

The table entries show the results of the ANOVA-Test. When two certainty maps are compared, only statistically significant results are shown. 'Sequence' refers to the sequence of the test map in the experiment.

D.F. : degrees of freedom; **SIGN.:** level of significance of F-Statistic

Table 2:
Response-Times of Park Selections Compared for Different Test Maps

This result appears to be counterintuitive at first, since darker value has been repeatedly suggested for the depiction of more certain information because it is perceived by map reader as being more prominent. Lighter value, on the contrary, is perceived as being less prominent (MacEachren, 1992; McGranaghan, 1993; van der Wel et al., 1994). It appears that this is true, if certainty information is depicted on printed paper, where colors are perceived by reflected light. On a CRT, however, where colors are perceived with emitted light the results might be reversed. Such change in perception has been already noted by Robinson et al. (1984).

Symbolization schemes for making fast siting decisions The ANOVA test was calculated between test maps to explore response times for the symbol schemes. Results are displayed in Table 2. An increase in map detail has differing effects on response times. Response times increase significantly when the number of attribute classes increase (comparing Map1 with Map3). Test subjects seem to need more time to mentally process the additional attribute classes. However, when the additional attribute classes include map certainty information, response times are either the same as or shorter than the one-class map. None of the differences in the response times between each certainty map and the one-class map are statistically significant. It would seem that map certainty information is understood as clarification rather than adding complexity to a map display. This result reiterates the need for additional testing.

Which symbolization scheme for the display of attribute certainty should be chosen? The result of the ANOVA-Test for the category 'all certainty maps' shows that the response times for all six certainty maps are not significantly different from each other. However, the results of the ANOVA-Test calculated between MapS and MapSi, and between MapSi and MapTi are significantly different. This suggests that either saturation or texture can be used to symbolize certainty information when decisions must be made quickly. If the symbol choice is saturation, then more pastel shades should be used to display the more certain information.

Symbolization schemes for making confident siting decisions Results of comparing subjects' confidence about their decisions are shown in Table 3. No significant differences were found when comparing Map1 with Map 3, nor when comparing Map1 with any of the certainty symbolization schemes. This implies that the decisions were made with confidence regardless of introducing additional information (i.e., that it was an easy decision to make in any case). However, comparisons between value and texture symbolization schemes do show significant differences in subject confidence. Subjects are overall more confident of decisions when certainty is symbolized by either lighter or darker value, than when symbolized by texture.

SUMMARY

This research demonstrates that inclusion of attribute certainty on thematic maps does modify spatial decision-making. Improvements in the number of correct decisions were observed when attribute detail is increased, either by additional classes or by including certainty information. Of the three tested symbolization schemes, value and texture were shown to improve the frequency of correct decisions. When correct decisions are the highest priority, lighter values should symbolize more certain information. When value is not available (if it has been used to symbolize other information on the map), finer texture should be applied instead.

TEST MAPS	MEAN RANK	CHI-SQUARE	D.F.	SIGN.
Map_1 and Map_3	1.56 / 1.44	0.8889	1	0.3456
Map_1 and Map_S	1.54 / 1.46	1.0000	1	0.3173
Map_1 and Map_{Si}	1.50 / 1.50	0.0000	1	1.0000
Map_1 and Map_V	1.54 / 1.46	1.2857	1	0.2568
Map_1 and Map_{Vi}	1.56 / 1.44	1.1429	1	0.2850
Map_1 and Map_T	1.46 / 1.54	0.8182	1	0.3657
Map_1 and Map_{Ti}	1.53 / 1.47	0.4000	1	0.5271
Map_V and Map_T	1.41 / 1.59	3.0000	1	**0.0833**
Map_{Vi} and Map_T	1.41 / 1.59	4.0000	1	**0.0339**
All Certainty Maps	*	4.7036	5	0.4531

The table entries show the results of the Friedman-Test. When two certainty maps are compared, only statistically significant results are shown.

D.F. : degrees of freedom; **SIGN.:** level of significance

* The mean ranks are: Map_{Vi}=3.31; Map_V=3.37; Map_S=3.38; Map_{Ti}=3.46; Map_{Si}=3.65; Map_T=3.84;

Table 3:

Confidence Level of Park Selections Compared for Different Test Maps

The most interesting results in this research were discovered for subject response times. One would expect that adding attribute information of any kind should slow down subject response times. Adding attribute classes had exactly this effect. However (and this is the interesting result) adding attribute certainty did not increase response times. No significant differences in response times were found in comparing one class maps with attribute certainty maps. This finding implies that map readers do not assimilate attribute certainty in the same way as they assimilate added map detail. Inclusion of certainty information appears to clarify the map patterns without requiring additional time to reach a decision. An experiment to observe response times for one-, two-, and three-class maps would be one way to confirm these results. Fastest response times were discovered for certainty maps showing saturation, thus if a fast decision is the highest priority, attribute certainty should be symbolized by more pastel colors.

Results indicate that symbolizing certainty by value gives subjects' greatest confidence in their decisions, although the decision task in this experiment was considered by subjects to be easy enough that high confidence was reported regardless of the symbolization scheme. In conjunction with the other findings, one can propose that attribute certainty can be symbolized most effectively using lighter values for more certain information, to ensure that correct decisions will be made more often. When value is not available, fine textures can show

attribute certainty almost as effectively. If quick decisions must be made, inclusion of attribute certainty will not impede response times, and use of saturation may in fact improve response times.

As a final point, one might consider the importance of empirical testing to establish guidelines for choosing effective map symbolization strategies. It is by means of rigorous subject testing that principles for map design may be formalized that were previously not known or not understood. In our work, the determination that introduction of certainty information may reduce the time required for spatial decision-making has been uncovered, and guidelines for symbol selection can be proposed. Additional testing can refine the results, of course. More important perhaps is the recognition that once the reasons for selecting a symbolization strategy have been formalized (in terms of correct decisions, or faster decisions), there are clear reasons for implementing such strategies as graphic defaults in mapping packages and decision support systems.

REFERENCES

Bertin, J. (1983). *Semiology of Graphics*. Madison: University of Wisconsin Press.

Buttenfield, B. P. (1991). Visualizing cartographic metadata. In Beard, Buttenfield, and Clapham, eds. *Visualization of Spatial Data Quality*. National Center for Geographic Information and Analysis Technical Paper 91-26, Santa Barbara, CA, pp. C17-C26.

Buttenfield, B. and D. Mark (1991). Expert systems in cartographic design. In Taylor ed. *Geographic Information Systems: The Computer and Contemporary Cartography*. Pergamon Press, Oxford, pp. 129-150.

Dent, B. (1996). *Cartography-Thematic Map Design*. Wm. C. Brown Publishers, Dubuque.

Federal Geographic Data Committee (1992). *Federal information processing standard. Publication 173 (Spatial Data Transfer Standard)*. U.S. Department of Commerce, Washington D.C.

Fischer, P. (1994a). Hearing the reliability in classified remotely sensed images. *Cartography and Geographic Information Systems*, 21(1):31-36.

Fischer, P. (1994b). Randomization and sound for the visualization of uncertain information. In Unwin and Hearnshaw eds. *Visualization in Geographic Information Systems*. Wiley, London, pp. 181-185.

Leitner, M. (1997). *The Impact of Data Quality Displays on Spatial Decision Support*. Unpublished doctoral dissertation, Department of Geography, State University of New York at Buffalo.

Leitner, M., and B. Buttenfield (1995). Acquisition of procedural cartographic knowledge by reverse engineering. *Cartography and Geographic Information Systems*, 22(3):232-241.

Leitner, M., and B. Buttenfield (1996). The impact of data quality displays on spatial decision support. *Proceedings GIS/LIS*, Denver, CO, pp. 882-894.

MacEachren, A. (1992). Visualizing uncertain information. *Cartographic Perspectives*, 13:8-19.

McGranaghan, M. (1993). A cartographic view of spatial data quality. *Cartographica*, 30(2):8-19.

Robinson, A., Sale, R., Morrison, J., and P. Muehrcke (1984). *Elements of Cartography*. John Wiley & Sons, New York.

Schweizer, D. M., and M. F. Goodchild. (1992). Data quality and choropleth maps: an experiment with the use of color. *Proceedings GIS/LIS*, San Jose, CA, pp. 686-699.

Van der Wel, F., Hootsmans, R., and F. Ormeling. (1994). Visualization of data quality. In MacEachren and Taylor, eds. *Visualization in Modern Cartography*. Elsevier, New York, pp. 313-331.

Weibel, R. (1995). Three essential building blocks for automated generalization. In Muller, Lagrange, and Weibel eds. *GIS and Generalization, Methodology and Practice*. London, and Bristol, Tenn., Taylor & Francis, pp. 56-69.

Zirbel, M. (1978). *Pattern Selection for Monochromatic Mapping of Nominal Areal Data*. Unpublished M.A. Thesis, Department of Geography, University of Kansas.

EXPLORING THE LIFE OF SCREEN OBJECTS

Sabine Timpf and Andrew Frank
Department of Geoinformation
Technical University Vienna, Austria
{timpf,frank}@geoinfo.tuwien.ac.at

ABSTRACT

This paper explores the metaphor 'screen objects are alive' for the purpose of zooming on geographic data at multiple levels of abstraction. In order to trace objects through multiple levels of detail we need to determine the areas these objects are associated to. This information is extracted from a partition tree. The paper first explains how to derive this partition tree. Then we define different lives of an object and show that they correspond to characteristic generalization operators. Analyzing life spans of (geo)graphical objects on screens throughout scale changes is crucial for the design of intelligent zooming mechanisms. It allows for the design of databases that are able to support highly dynamic user interaction in complex visualization tools.

1. MOTIVATION

Humans perceive, conceptualize and deal with the world at multiple levels of detail (Marr 1982, Minsky 1985). The need for a multilevel and multiperspective approach for geographic visualization is recognized in the Geographic Information society (Buttenfield and Delotto, 1989; MacEachren 1995). However, solutions how to handle multiple levels of detail in data structures or in user interface tasks like zooming are still missing. A good metaphor facilitates to find structures and operations for multiple levels of detail. This paper explores the metaphor 'screen objects are alive' for the purpose of zooming on geographic data at multiple levels of abstraction.

The metaphor of life is a new metaphor for the operation of zooming. In the metaphor 'screen objects are alive' we consider the dynamic process of zooming the contents of a display. Objects are born when they are first represented on the screen. They die when they disappear from the screen. These changes (being born and dying) as well as other transformations occurring during scale change have been called 'catastrophic changes' (Mueller, 1991).

We assume that the database contains data in multiple levels of resolution. When zooming in the object in focus appears on the screen, then grows larger and larger, perhaps splits into several objects and finally is too large to be in focus. Another smaller object, that was part of the original object, now becomes the focus. In the other direction, zooming out, the object shrinks, becomes smaller and smaller until it disappears. Our objects change on the screen because we change their level of resolution when we zoom in and out.

In this paper we analyze how screen objects behave when zooming. We use a scanned map series from former East Germany as test data. The representations of the objects change with scale. In a GIS, there are other operations that can cause screen objects to change, e.g., reselection of topic, temporal change, or panning. In this work we are only concerned with the operation zooming, while theme, time and space are fixed. This excludes specifically multi-topic zooming from the scope of this paper (for works on thematic zooming see Volta 1992). The area of space that is radially the farthest from the focus of the zoom disappears when zooming in. This is, because we also consider the display window to be fixed in its dimensions.

Analyzing life spans of (geo)graphical objects on screens throughout scale changes is crucial for the design of intelligent zooming mechanisms (Bjorke and Aasgard, 1990; Timpf, to appear; Timpf and Frank, 1995). It allows for the design of databases that are able to support highly dynamic user interaction in complex visualization tools (Goodchild 1990).

The remainder of this paper is organized as follows: section two explains how we partition space to create objects and traces objects over three scales. Section three presents the partition tree we use to store object changes. Section four explains that metaphors help us in structuring and understanding our area of research. It also examines the results of section three in the light of the metaphor 'screen objects live'. Section five gives conclusions and proposes future work.

2. WHAT ARE OUR OBJECTS?

In this paper we analyze and describe objects that were captured from a series of maps from prior East Germany. The maps have been scanned with non-professional equipment and saved as TIF files. Our examples are drawn from the areas of Alsleben/Saale and Berneburg/Saale. The map scales considered are 1:10 000, 1:25 000, 1:50 000, and 1:100 000. The last three were created with the same symbolization scheme.

In this section, we assume that our screen objects show the same behavior and structure as map objects. Although this is not a requirement of our model, it is the only practicable way to observe objects over several scales. We first explain how to subdivide map space and how to derive objects from that process.

We then trace map objects over three scales and create their respective partition trees.

2.1 Map space is a container

We regard map space as a container, that contains more containers. This hierarchical arrangement of containers (Fig. 1) can be represented as a tree and corresponds to the vertical zooming hierarchy.

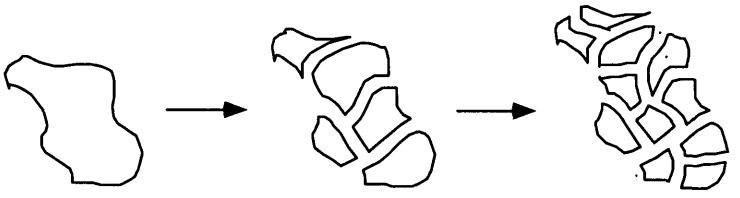


Fig. 1: Hierarchy of containers

At each level of the tree the set of containers is a complete partitioning of map space. One possibility for a complete partitioning of space is the subdivision into administrative units. Administrative units are arbitrary partitions of space, they do not reflect the underlying structure of space. From a visual point of view, those lines that are black and broad give a first subdivision of space. Lagrange (1994) and Bannert (Bannert, 1996) have proposed a division of space with the help of the street network. This idea is taken up and extended here: We divide space by the hydrographic network, the train network, and the street network. We start with the network that is preserved the longest in each of the three mentioned classes (Fig. 2). The density of the network grows with scale when more rivers, railways, and streets are added to the existing network. The method requires a consistent division of space over all scales. This means that the lower levels need to be completely included in the higher levels. When using real maps this often presents a problem.

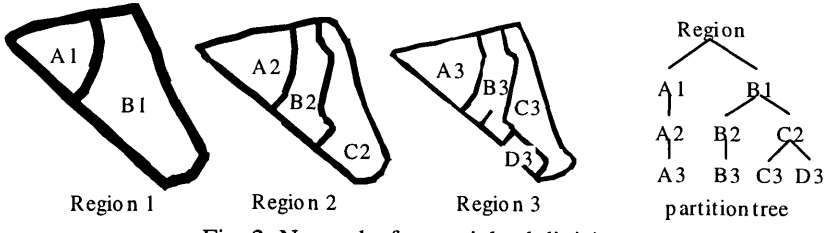


Fig. 2: Networks for spatial subdivision

The space between the lines of either network is considered a container. In the example (Fig. 2) we picked the same region in three different scales; the scale grows from left to right. In this particular case only the street network subdivides the space. Region 1 is a container that contains two areas A1 and B1. Region 2 contains three areas A2, B2 and C2. Areas B2 and C2 in region 2

correspond to area B1 in region 1. Region 3 contains four areas A3, B3, C3, and D3. Areas C3 and D3 in region 3 correspond to area C2 in region 2. In this example the partitions are consistent and can be represented by a tree (on the right in Fig. 2).

2.2 Contents of a container on several levels

The containers as defined above can either contain another partition of space based on the use of the area (e.g., industrial area), or objects like single houses and symbols, or both. In the following example (Fig. 3 through 5) four different areas have been identified. They are house block area, residential area, garden area, and non-designated area. The last three can contain objects like houses and symbols. Areas are determined either through a color change or through an existing boundary, that is not a street. If an area contains a street that divides the area into two or more separate areas, new containers are created. This means that there is another level in the partition tree.

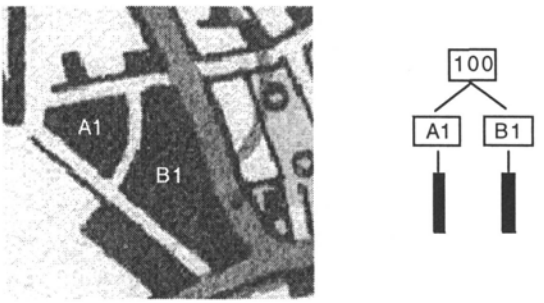


Fig. 3: Contents of a container (region 1 of figure 2)

The content of a container can also be represented by a tree. E.g., in figure 3 a high level container (called 100) contains two lower levels containers A1 and B1. Both A1 and B1 contain just one area, which is a house block area. The same region with more detail contains similar containers A2, B2, and C2.

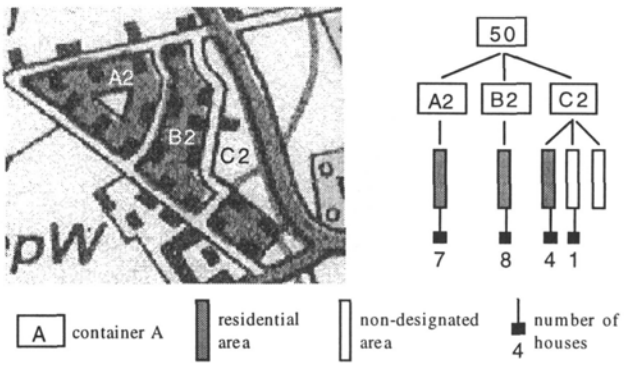


Fig. 4: Contents of a container (region 2 of figure 2)

The containers A1 and A2 cover the same area, whereas the containers B2 and C2 form a partition of B1. All of these containers contain other objects. E.g, the container C2 includes three areas: one residential area with four houses, one non-designated area with one house and one empty non-designated area.

Figure 5 shows again the same area with more detail than figures 4 and 3. There is a new container D3 in this example, that together with container C3 forms a partition of C2. In this last example we have omitted the tree description of container A3 for clarity.

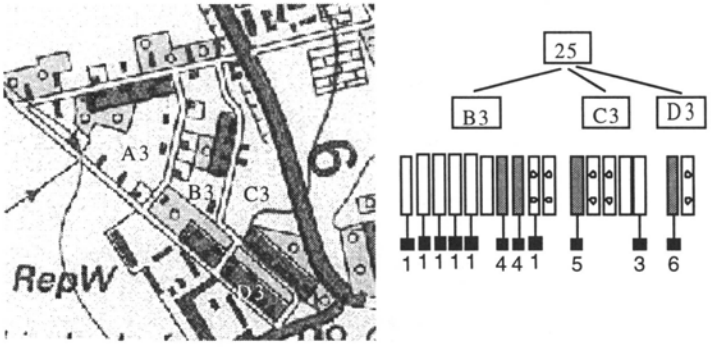


Fig. 5: Contents of a container (region 3 of figure 2)

The container C3 in figure 5 includes five areas: one residential area with five houses, two garden areas and two non-designated areas with one area containing three houses. The container D3 in figure 5 includes one residential area with six houses and a garden area.

These examples have shown that a consistent partitioning of space is possible. The results of this partitioning are three partition trees shown in the right hand side of figures 3, 4, and 5. In the next section we fit together the contents of all three partition trees according to their levels.

3. PARTITION TREE

In this section we determine how the combined partition tree looks like. The combined partition tree is necessary to trace objects over levels of detail and thus determine how they lead their live. The life of objects cannot be determined if the partitioning of space is not consistent or if some levels have ambiguous links to other levels. We have chosen an example where we can determine the life of objects. In figure 6 three partition trees are shown for container B1 and its partitions in regions 1, 2, and 3 respectively.

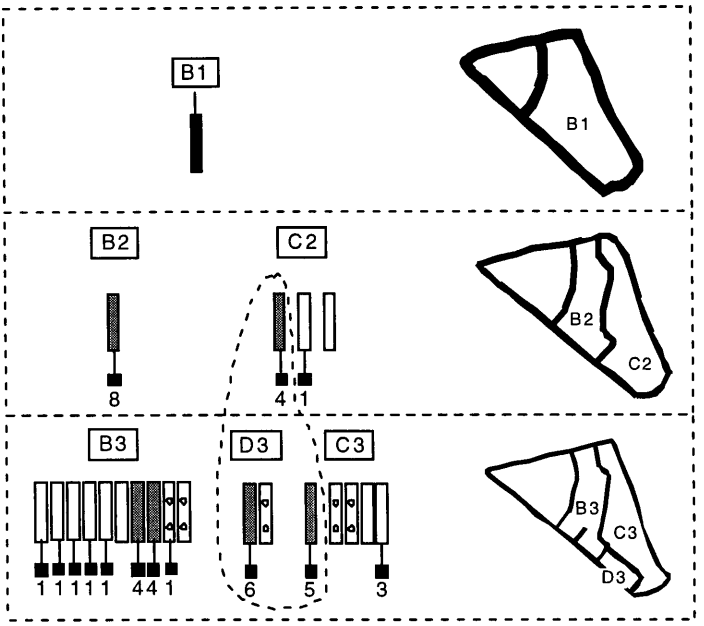


Fig. 6: Combined partition tree and corresponding regions

Figure 6 shows the partition tree in detail with its two levels and the corresponding regions. We already know from the partition tree in figure 2 that the container C2 in the upper level is subdivided into two containers C3 and D3 in the lower level. In the more detailed partition tree, we can see that the subdivision is not as clear. The residential area in C2 is split into three areas in the containers D3 and C3.

There are objects on each map that do not partition space in our model. Those are for example isolines, power lines, and embankments. These do not build containers (with exceptions of course, which we do not consider here). Therefore we disregard in this paper isolines, power lines, embankments, and also labels. In the last case there are already working algorithms that can be applied to an otherwise finished product (Freemann 1991).

4. THE LIFE OF SCREEN OBJECTS

The partition tree in the previous section is the basis for our model of a life of objects. Objects from different levels of resolution are related to each other through the partition tree. The levels of the partition tree reflect the scale of the screen objects. While zooming, the scale continuously changes and with the scale the levels of detail change. We use the metaphor 'screen objects live when zooming' to express this relationship between scale change and time change.

4.1 Why use a Metaphor?

The metaphor of life is a new metaphor for the operation of zooming. In the metaphor 'screen objects live' we consider the dynamic process of zooming the contents of a display and observe how objects change in this process. Objects are born when they are first represented on the screen. They die when they disappear from the screen. Our objects change on the screen because we change their level of resolution when we zoom in and out. The time that passes during zooming (either in or out) is the time that our objects live. What is different from the life we know is, that we can move forward and backward in time by zooming in and then out again or the other way around.

Metaphors allow us to understand (sometimes only partially) one thing in terms of another. This does not mean that the two things are the same or even similar (Lakoff and Johnson, 1980). Metaphors are mappings of structure or behavior from a source domain to a target domain. For example the metaphor 'life is a journey' applies the notions of a journey to the notion of life. This is reflected in expressions such as 'he is off to a good start' or 'its time to get on'.

Jackson has identified the need to understand the user interface operations of zooming and panning more deeply (Jackson, 1990) . Since most of our fundamental concepts are organized in terms of one or more spatialization metaphors. (Lakoff and Johnson 1980, p.17), we think that metaphors will help to understand the operation zooming.

Cartographers still struggle to understand the notions of scale and resolution (Kuhn, 1991). We think that the source domain of 'life' can be metaphorically applied to the target domain of 'zooming'. This helps to understand the target domain and sheds light on the problem of scale.

When we interpret screens as dynamic views on data and not as static maps, we allow objects on the screen to change. In maps, things do not change. They are static representations of a state of the world. As Kuhn (1991) has argued convincingly, the metaphor 'displays are maps' is restricting the possibilities we have with current GIS systems (Moellering 1984, Robertson 1988). He proposes to use instead the metaphor 'displays are views'. Views are dynamic representations and things may change in views. We go further in this metaphoric chain and say 'changing things are alive'.

A similar metaphor has already been introduced by Buttenfield (Buttenfield, 1991) and pursued by Leitner (Leitner and Buttenfield, 1995). They talk about the behavior of cartographic objects over several scales. The goal of their study is to formalize rules about the behavior of cartographic objects. Our aim is to describe the behavior of screen objects with the help of a structuring metaphor.

4.2 Different Lives

The hypothesis for zooming out is that the object shrinks (shows less detail) until it is in its most abstracted state or prototype. The hypothesis for zooming in is that the object grows (shows more detail) until every detail known on this level in the database is displayed. Then, in both cases, a transformation (or catastrophic change) takes place and the same process repeats itself again.

We have found that a distinction between four different lives is necessary. Let us assume that we traced an object over several levels of detail starting from the most detailed representation. Let us further assume that the object is symbolized on the second level. On the third level the object disappears (see Fig.7).

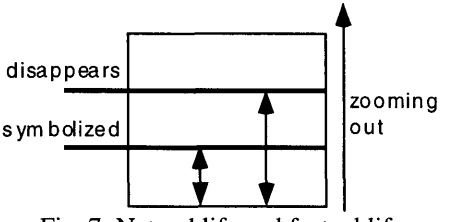


Fig. 7: Natural life and factual life

Natural life is the life a screen object would have if there were no cartographic rules. It would disappear the moment it is too small to be represented. In our example this would be the case after the first level. But the object is symbolized and thus its factual life is longer. The **factual life is** the life the object lives on screens thanks to the interpretation of a cartographer, who judges the object too important to be dropped. There are some objects that, by definition, will have different factual and natural lives (e.g. official buildings, important routes between cities etc.). Objects that are displayed as symbols are always prolonging their natural life. A symbolized object is leading an artificial life.

We also make a distinction between objects that live a short life and those that live a long life. The **short life** of an object lasts from its appearance to its disappearance on the screen as a *separate* object. In this life, the object dies when it is merged with others (zooming out). It also dies when it is split up into several objects (zooming in). When it is symbolized, it lives on (artificial life). The **long life** of an object is from the moment it appears up to the moment its last trace disappears from the screen. In the meantime the object can change several times through simplification, aggregation, symbolization, splitting, geometrization etc.

The definition of several lives of objects has shown that each life supports a different characteristic of screen objects in several levels of detail. It is part of future work to formally define these changes and their corresponding operations.

5. CONCLUSIONS AND FUTURE WORKS

In this paper we applied the metaphor of living = zooming to screen objects. We had to use existing maps in several scales as test data. We partitioned map space according to existing areas and built a partition tree for a map region in several levels of detail. With the help of the partition tree, we analyzed life spans of (geo)graphic objects. We found that several characteristics of screen objects can be captured by the definition of four different lives. These characteristics shed light on the problem of generalization but they also help to define and understand tools (especially zooming) for multi-resolution databases.

One important result of this study is that the partition trees need to be consistent in order to define the life of an object. This is often a problem with existing maps, which have not been created from the same scale and by the same person. It is necessary to find consistent test data, so that our study can be continued.

Future work in our research consists of formally defining partition trees and combining them into trees that span all levels of an object's life. After that it is necessary to formalize the currently defined lives and to verify the hypothesis that each of these lives supports a different generalization operation.

REFERENCES

Birgit Bannert, internal report: Fachdatenintegration in ATKIS für das Umweltinformationssystem Baden-Württemberg, University of Hannover, 1996.

Bjørke, J. T. and R. Aasgaard (1990). Cartographic Zoom. Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland; July 1990, IGU.

Buttenfield, B. P. and J. Delotto (1989). Multiple Representations: Report on the Specialist Meeting, National Center for Geographic Information and Analysis; Santa Barbara, CA.

Buttenfield, B. P., C. R. Weber, et al. (1991). How does a cartographic object behave? Computer inventory of topographic maps. GIS/LIS, Atlanta.

Freeman, H. "Computer name placement." In *Geographical Information Systems: principles and applications*, ed. Maguire, David J., Goodchild, Michael F., and Rhind, David W. 445-456. 1. Essex: Longman Scientific & Technical, 1991.

Goodchild, M. F. (1990). A geographical perspective on spatial data models. GIS Design Models and Functionality, Leicester, Midlands Regional Research Laboratory.

Jackson, J.P. "Developing an effective human interface for geographical information systems using metaphors." In *ACSM/ASPRS Annual Convention in Denver, CO; March 18-23, 1990*, 117-125, 1990.

Kuhn, W. (1991). Are Displays Maps or Views? ACSM-ASPRS Auto-Carto 10, Baltimore, Maryland, American Congress on Surveying and Mapping.

Lagrange, J.P. and Ruas, A. "Data & Knowledge Modelling for Generalisation". In SDH 94, Symposium on Spatial Data Handling, Edinburgh, Scotland. Ed. T. Waugh and R.G. Healey, 1994.

Lakoff, G., and Johnson, M. *Metaphors We Live By*. Chicago: University of Chicago Press, 1980.

Leitner, Michael, and Buttenfield, Barbara P. "Acquisition of Procedural Cartographic Knowledge by Reverse Engineering." *Cartography and Geographic Information Systems* , 22 (3): 232-241, 1995.

MacEachren, A. M. (1995). How maps work. New York, The Guilford Press.

Marr, D. (1982). Vision. New York, NY, W.H. Freeman.

Minsky, M. (1985). The Society of Mind. New York, Simon & Schuster.

Moellering, H. (1983). Designing Interactive Cartographic Systems Using the Concepts of Real and Virtual Maps. Sixth International Symposium on Automated Cartography, Ottawa, Ontario, Canada.

Muller, J. C. "Generalization of spatial databases." In *Geographical Information Systems: principles and applications*, ed. Maguire, David J., Goodchild, Michael F., and Rhind, David W. 457-475. 1. Essex: Longman Scientific & Technical, 1991.

Robertson, P. K. (1988). Choosing Data Representations for the Effective Visualisation of Spatial Data. Third International Symposium on Spatial Data Handling, Sydney, Australia, International Geographical Union.

Timpf, S. and A. U. Frank (1995). A Multi-Scale Dag for Cartographic Objects. ACSM/ASPRS, Charlotte, NC.

Timpf, S. (to app). Cartographic objects in a multi-scale data structure. Geographic Information Research: Bridging the Atlantic. M. Craglia and H. Couclelis.

Volta, Gary. "Interaction with attribute data in Geographic Information Systems: A model for categorical coverages." Master of Science, University of Maine, USA, 1992.

SHAPE ANALYSIS IN GIS

Elizabeth A. Wentz
Department of Geography
The Pennsylvania State University
University Park, PA 16801 USA
wentz@geog.psu.edu

ABSTRACT

The essential objectives in the analysis and subsequent understanding of geographical phenomena involves searching for spatial patterns, followed by evaluating possible causes and effects of patterns, and predicting future patterns. On a human level, the visual identification and comparison of areal shapes is a fundamental and integral component of this process. Scientists and practitioners across a variety of fields have recently turned to GIS technology as a central component to manage and analyze observational data. Despite a rich heritage of techniques for describing and analyzing shape within geography as well as other fields, such capabilities generally remain quite primitive within GIS. A general purpose shape analysis capability that even approaches the level of sophistication of other current GIS capabilities is still not part of the GIS toolkit. This paper examines the need for a shape analysis capability and examines potential approaches to extend GIS for handling more powerful shape analysis techniques.

INTRODUCTION

Imagine that you are a field biologist interested in habitat areas for primates in Costa Rica. Through various techniques, including radio telemetry, field mapping, and a GIS, you have identified two regions that are possible habitats for large monkey troops. The GIS component consists of using overlay techniques, distance measurements, area calculations, and a shape function, to extract common physical variables including: slope, vegetation type, distance to streams, area, and an irregular lobed shape. One of your goals is to protect the species; consequently, your objective is to understand how and why these regions are suitable habitats so that you can search for other similar regions. You extract from your GIS all other regions with the identical physical conditions and then ask additional questions regarding the shape of these areas such as: 1) How has departure from a specified "ideal" shape and size changed the behavior of the different troops? 2) What is the juxtaposition of these particular shapes with other land characteristics (e.g., nearness to water)? 3) Can changes to the shape of habitats be projected into the future based on the underlying processes of soils, geology, and current vegetation? With existing GIS software, these questions would be difficult to answer because they require a shape analysis capability. Extracting all regions with similar irregular lobed shapes would

require a manual, visual examination of each polygon to determine if they match your idea of a habitat shape.

The gains in ecology from a shape measure are evident in the analysis of habitat delineation (Eason 1992; Gutzwiller and Anderson 1992). Eason (1992) claims that scarcity of resources is forcing the optimization of territories and this requires knowledge beyond the size, vegetation, and topography of a given region. It was once believed that a round territory was ideal within a homogenous region, yet these assumptions are beginning to be proved incorrect. The geometry of the area is now considered a critical factor in defining and analyzing habitat regions.

There is a continued interest and need for shape analysis in many fields of study; the ecology example given above being only one of many potential applications. In other contexts, scientists are concerned about characterizing and comparing spatial shape when studying the effects of urban growth, analyzing the effects of earthquakes, determining the impact of deforestation, and measuring the parameters of drainage basins. Specifically, urban geographers have used shape indices to quantify the shape of political districts to explain spatial patterns and studied the organization of transportation to assess the changing shape of cities over time (Gibbs 1961; Simons 1974; Lo 1980; Austin 1981; Rhind, Armstrong et al. 1988). As a result of these applications and many more, numerous attempts have been made to quantify shape for the identification and subsequent comparison of geographic regions (Boyce and Clark 1964; Lee and Sallee 1970; Moellering and Rayner 1982).

Given the impact that identifying shapes has in understanding spatial processes, and the use of GIS in geographical research, improved GIS analytical capabilities includes the quantitative characterization of the form of regions. Visually identifying and comparing regions on the basis of shape is easy and intuitive for humans to do. Our visual/cognitive system is well-attuned to this kind of task. Given the increasing frequency of large volumes of data in GIS and the need for analysis over large geographical areas, make the visual approach impractical. The goal of this paper is to suggest a method for improving the ability to compare the shape of regions in a GIS environment that is less dependent on direct human intervention.

WHAT IS SHAPE ANALYSIS?

Pattern identification is critical in understanding spatial relations, and pattern and shape are closely linked. Consequently, for clarity in this research, an explicit definition of pattern and shape is given. This discussion provides the basis for a detailed description of shape analysis and definition of shape indices in the context of studying geographic scale phenomena.

Pattern can be described as the organization of phenomena in geographic space that has taken on some "specific regularity, which in turn is taken as a sign of the workings of a regular process" (Ebdon 1988). Shape on the other

hand is more basic; shape describes the geometric form of individual spatial objects (MacEachren 1985). An object in geographical terms is defined as a foreground placed on some type of background. Examples include lakes, common soil types, political boundaries, or any area defined to be homogeneous with regard to some characteristic or combination of characteristics. The regular or irregular organization and juxtaposition of these individual areas provides the building blocks for describing spatial pattern. Pattern alone is highly complex, but the measurement of shape in the context of pattern provides a mechanism to simplify pattern into basic units.

Shape analysis is the process of building fundamental units for identifying and describing patterns in the landscape. The requirements for this process are to describe shape, including a distinction between regular and irregular shapes, and to answer questions regarding shape (Pavlidis 1978; Moellering and Rayner 1982; Ehler, Cowen et al. 1996; Xia 1996). Describing shape involves identifying the outside boundary of an object in space. Another component to the description of shape involves the description of both regular and irregular shapes. Regular geometric shapes, such as circles, squares, and triangles can be described simply. Irregular shapes, however, can be highly complex with infinite variations and are more likely to appear in a geographical context. The second step in the process involves addressing shape comparisons. Distinguishing between areas that have holes (such as an island within a lake) and areas with different edge roughness, where the overall geometric configuration does not vary, is part of the geographical analysis.

An index to describe shape and allow for comparisons must meet several specific criteria. MacEachren states that "the first [criterion] is to develop a measure of shape uniqueness by which any shape can be distinguished from all other shapes and similar shapes result in similar descriptions" (MacEachren 1985). The list of criteria for a shape index for this research are: 1) each unique shape be represented with a unique number; 2) independent under translation, rotation, size, and scale change; 3) match human intuition; 4) deals with regions that contain holes; 5) easy to calculate and interpret the results. The ideal shape index would meet all these criteria.

The primary reason that shape analysis is not currently part of the GIS toolkit is that no single satisfactory method has been developed (Lee and Sallee 1970; Ehler, Cowen et al. 1996). Nevertheless, numerous indices have been suggested varying from simple area and perimeter calculations, to complex indices using sophisticated mathematical functions. The evidence that no method exists is suggested in a review of "successful" implementations of shape indices. For example, in geography Frolov (1974/5) and later, MacEachren (1985) summarized techniques to measure the compactness of regions. Frolov focuses on the history of the various approaches, but MacEachren summarized and categorized the indices. MacEachren provides a systematic comparisons of the various methods but never suggested any single approach as the best method for measuring shape.

RESEARCH ON SHAPE IN OTHER FIELDS

The field of geography and the geographical context is only one of a range of areas where shape is important. Computer science, mathematics, computational geometry, statistics, and cognitive science also participate in the search for shape description and representation. Nevertheless, the application, and therefore the goals and definitions in each discipline, are different. This section briefly describes the approach taken by each field and suggests possible commonalties and contributions.

Within computer science, shape analysis is studied in the context of graphics and visualization including three dimensional graphics, animation, and character recognition. Much of the research on shape analysis within these areas has used a decomposition and construction approach. For example, Marr (1982) derives a technique to represent shape with collections of small cubes packed together in arrangements to approximate the shape of the given object. Marr's techniques, as are others for similar applications, are based on the need to recognize and represent shapes graphically rather than forming a unique descriptor that can be used to compare them. Consequently, the techniques developed by Marr and others in computer science have the objective of shape identification and depiction.

The literature available within mathematics, computational geometry, and statistics is extensive (Smith 1966; Davis 1977; Lord and Wilson 1984; Grenander 1996). Unlike computer science, the research in these disciplines is not directed by an application context. Instead, shape analysis is viewed more as an interesting problem to solve in the abstract, such as use of Delaunay triangles or convex hulls (Preparata and Shamos 1985). These methods do not present a method for extracting a single number to describe shapes, which can subsequently be used as a shape measure in a geographic context.

Much of the understanding of what shape is, regardless of the computational or mathematical application, is derived from the human visual sense. In a context different from the development of algorithms and formulas for shape descriptions, understanding of the visual aspect of shape comes from cognitive science through shape recognition, the creation of shape categories, and the language of shape. Landau et al. (1988) conclude that for perception and categorization of objects, shape recognition (other topics considered were size and texture) is significant for children who are learning words. Consequently, shape recognition is a skill that people acquire in the first few years of life. Without any additional proof, it is logical to assume that this skill is carried through life as a natural and intuitive process for identifying objects. This method of examining the world can be applied to the identification and categorization of regions with similar geometric form in the human and physical landscape.

The objectives for shape analysis from computer science, mathematics, statistics, and cognitive science are different. Nevertheless, there are overlapping agendas that contribute to meeting goals for shape analysis in GIS context. Shape analysis can be extended from the creation of a shape index to include

shape extraction, as derived from computer science. This technique could be applied to remote sensing applications, where the objective is to classify homogeneous regions, such as the extent of lava flows (Xia 1996). Mathematics and statistics strive for a method to define and represent shape, which in the context of geographical analysis could be then developed into an index. Cognitive science contributes to the human-oriented needs for a shape capability through the creation of categories of similar shapes. In a GIS context, the index represents a method for comparing geographic regions so that an oblong object can be categorized with another oblong object, and jointly categorized as "potentially similar wildlife habitats". As a direct consequence of the numerous approaches to shape, there are many possible types of shape indices. In order to evaluate these as possible shape indices in the context of GIS, they can be categorized and assessed based on similar qualities.

CATEGORIES OF SHAPE INDICES

Applying the approaches from research in other fields combined with a geographic definition of shape analysis, three general categories of existing shape indices can be identified. These categories are in contrast to the ones suggested by Pavlidis (1978), or MacEachren (1985). The Pavlidis categories were limited to measures for shape recognition (e.g., character recognition). He defined two categories based on whether they examine only the boundary or the whole area and whether they describe objects based on scalar measurements or through structural descriptions. MacEachren evaluated only measures that were based on compactness, which have been combined into one category for this research and will be described in this section. The categories presented here, on the other hand, include the types of indices Pavlidis describes for shape recognition and the compactness measures described by MacEachren plus a broader range of indices. These categories are compactness measures, boundary measures, and components measures. This section defines each category and evaluates generally how well the indices in the category meet the criteria for an index.

Compactness Measures

In a geographic context, shape is often characterized through a compactness indicator, which describes the form of a given region based on how far it deviates from a specified norm (e.g., circle, square, or triangle). The regular shape (normally a circle) is given the value 1.0 and less compact regions are typically less than 1.0 (e.g., 0.54343), where the smaller the number it is, the further it is from a non-circular region. The method for calculating this number utilizes one or more of the geometric parameters of the region being measured, such as area or perimeter. The parameters used and the mathematical equation depend on the feature of shape being measured, such as elongation or indentation.

With compactness measures, differently shaped regions produce different numbers, consistent with the criteria for a shape measure. Nevertheless, these measures do not represent true measures for shape because the number depends on the scale and size of the object, which does not meet the criteria defined in this research. Compactness indices are useful, however, in some contexts and

are important to geographers because "compactness is often considered to be indicative of homogeneity within units. The more compact a unit is, the shorter the average distance between any two locations, therefore, the more similar characteristics of those locations are likely to be" (MacEachren 1985).

Boundary Measures

Boundary measures describe shapes by outlining the perimeter of a region. The index is assigned based on the technique, mathematical or otherwise, used to outline it. Several of these approaches have been applied to geographic examples, but many have not. One of the critical limitations with these indices is that they do not take holes in the region into account in an explicit manner. Some of the boundary measures that have been applied in a geographic applications include Fourier series, fractal analysis, Hough transforms, and Freeman chain codes.

Although several of theses indices are independent of rotation, scale, and translation, and provides a technique to regenerate the region, they do not provide a single index that can be used to compare the shapes of regions. For example, the Dual Axis Fourier Shape Analysis generates several numbers that, when combined, form a representation of shape (Moellering and Rayner 1982). The numbers, however, are difficult to calculate because the method requires that the points of the polygon be digitized at fixed intervals. In GIS applications, the stored digital coordinate data usually do not meet this restriction and re-sampling would mean a time consuming extra step. Other boundary measures do exist that address these issues, but they also have limitations, such as producing an index that is complex to interpret and not identifying regions with holes.

Component Measures

The components measures is the final category of shape indices. These measures describe the form of a region by deconstructing the region into combinations of regular shapes such as squares, circles, and triangles, as suggested in the computer science literature. The number and type of regular shapes, and possibly other parameters, become the index. This is necessary if comparisons are to be made between regions because it is conceivable that two different regions could be made up of the same combination of regular shapes, but because of different organizations, the visual shape of the regions would be different.

The strength of the components measure is that it breaks irregular shapes into regular shapes, which can then be numerically defined. There are, however, many weaknesses. The primary weaknesses of the components measures are they generally do not maintain topology and consequently they do not retain the same characteristics under translation and rotation. Additionally, complex regions result in complex indices, which are difficult to interpret and tend to oversimplify the original region.

Summary

The indices just described are diverse and measure different elements of shape. Compactness measures do not distinguish between regions with holes and do not indicate edge roughness. Boundary measures are well suited for measuring edge roughness, but little can be done for identifying holes in regions and they are often complex to calculate. The components measures represent a descriptive measure for shape rather than a quantitative (thus comparable) measure for shape. This paper has identified the need for a shape analysis capability in a GIS context, and found that no suitable measure in its current form exists. Nevertheless, there is a potential approach that will allow for a more powerful shape analysis technique.

THE OUTLOOK FOR SHAPE ANALYSIS IN GIS

The basis for this research suggests that compactness measures, or any of the other indices for that matter, alone do not measure enough aspects of shape. The compactness measures, although they match our defined criteria, are not suitable measures for shape because two similar shapes can have different numerical representations and two different shapes can have similar numbers. For example, Figure 1 shows two shapes (regions A and B) that have similar numbers based on the area-perimeter compactness measure (0.489 and 0.491), but are not similar in shape. In the same figure, a third region is identified (region C) that is similar in shape to region B, but the shape index (0.429) is not as close as the index for region A. Consequently, evaluating the shape of these regions based on the compactness measure does not effectively measure shape. To examine two shapes that are different in area and perimeter, Figure 2 highlights two regions that have a similar index, but are quite obviously different in shape. Similar examples for boundary measures and components measures can be made.

Figure 1

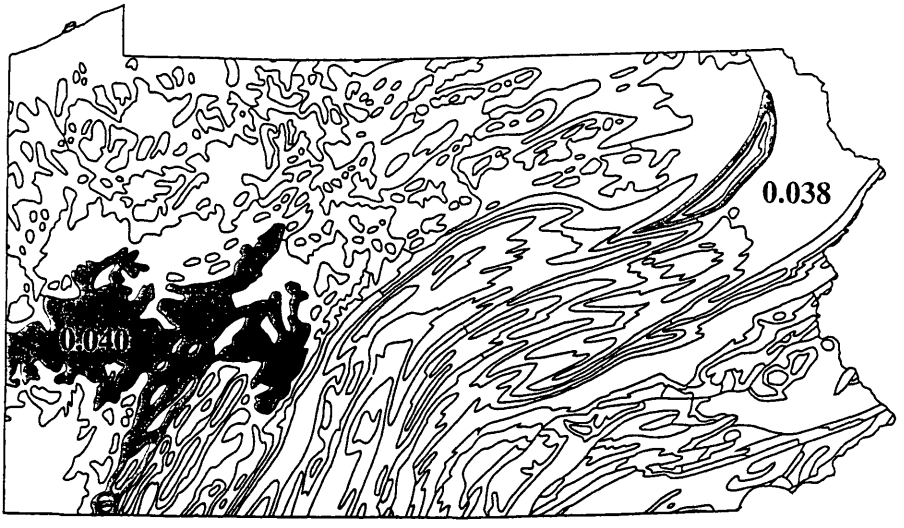


Figure 2

The problem with evaluating the indices based on this list of criteria is that the original definition of a shape index is too simplistic. Understanding the geometric form of a region may require something that does not simply match human intuition, consequently these criteria are too narrowly and defined. Ease of calculation, another one of the defined criteria, is important in the implementation phase, but should not be a deterrent if the interpretation is easy and it describes geometric form.

Intuitively, shape seems like a simple concept, although in actuality, the human visual/perceptual mechanism for distinguishing and recognizing shape is complex. Consequently, in a mathematical context, shape is also complex. Unlike measurements for area or perimeter, the definition of shape is based on linguistic expressions and human intuition. Presently, mathematics and statistics are not in a position where shape can be reduced to a single number. Due to complexity of the concept, a single numerical representation of shape may be impossible to achieve.

The results from the present assessment of shape analysis in GIS could be taken in several directions. One possibility is to redefine shape. This research is proposing a new means of characterizing shape involving deconstructing shape into components, where each component can be represented with a number. This method is similar to the way color (also visually and conceptually simple) is separated into hue, saturation, and value. In the case of shape, the properties proposed are edge roughness, compactness, and geometric form. Existing indices, from each of the three categories of existing shape indices, could be applied to quantify the different properties of shape. Boundary measures are the best at assessing edge complexity, compactness measures, as the name implies, evaluates most effectively the closeness to a compact form such as a circle, and

geometric form (including whether the region contains holes) is best evaluated with the components measures. For example, fractals appear suitable to measuring edge roughness, area-perimeter measures could measure compactness, and a Triangulated Irregular Network (TIN) could represent geometric form. Each region could have three shape indices assigned to it, each based on a different property. The deconstruction of shape into these constituent components is in its early stages. The next phase of the research is to re-examine the indices to determine which would best represent a particular property of shape. Viewing each index as representing a property of shape has given a new definition to shape.

CONCLUSION

A new definition of shape could allow for several different types of analysis in a GIS context to take place. Extraction and comparison of regions with "similar shape" could utilize indices individually or in conjunction with one another. For example, in the case of a solid waste landfill siting project, compactness and geometric form are important properties. Suitable landfills are not long and skinny and do not contain holes (e.g., a new landfill would not be situated with a residential development contained within it). Compactness measures and a measure describing geometric form could identify a suitable region. Edge roughness, however, is not critical for the siting of a landfill. In the case of a habitat delineation project, edge roughness is the important component. It has been suggested that certain species prefer habitat boundaries that are non-linear because they provide protection against predators that smooth boundaries do not provide.

Using existing measures to identify distinct properties of shape rather than expect any single measure to capture all aspect of shape is a better approach because these individual properties are important in themselves. In this way, it may be possible to identify similar regions that may not be classified as similar had they been evaluated visually. For example, two shapes with similar edge roughness may be classified "similar", even though their geometric form or compactness may be quite different. Similar processes, for example in geomorphology, could be at work. Holes in regions along with different areas or perimeters may mask the similarity of geometric form that compactness indices may extract.

This research presented here fits into a broader research project that investigates a theoretical approach to advancing GIS analytical capabilities. This theoretical framework suggests that improving the analytical capabilities of GIS requires more than simply the design and implementation of advanced numerical procedures. Rather, the research suggests alternative uses of computer technology that enhance human senses in place of replicating existing manual techniques. It is expected that future research will involve exploring cognitive approaches to shape analysis within the suggested framework.

REFERENCES

Austin, R. F. (1981). The Shape of West Malaysia's Districts. *Area* .

Boyce, R. B. and W. A. V. Clark (1964). The Concept of Shape in Geography. *Geographical Review* . 561-72.

Davis, L. S. (1977). Understanding Shape: Angles and Sides. *IEEE Transactions on Computers* **26**(3): 236-242.

Eason, P. (1992). Optimization of territory shape in heterogeneous habitats: a field study of the red-capped cardinal (Paroaria gularis). *Journal of Animal Ecology* **61**: 411-424.

Ebdon, D. (1988). *Statistics in Geography* . Worcester, Billing and Sons Ltd.

Ehler, G. B., D. J. Cowen, et al. (1996). Development of a shape fitting tool for site evaluation. *Spatial Data Handling* **1**: 4A.1-4A.12.

Frolov, Y. (1974/5). Measuring the Shape of Geographical Phenomena: A History of the Issue. *Soviet Geography: Review and Translation* . 676-87.

Gibbs, J. (1961). A Method for Comparing the Spatial Shapes of Urban Units. *Urban Research Methods* Ed. J. Gibbs. Princeton, Van Nostrand Co.

Grenander, U. (1996). *Elements of pattern theory.* Baltimore, John Hopkins University Press.

Gutzwiller, K. J. and S. H. Anderson (1992). Interception of moving organisms: influences of patch shape, size, and orientation on community structure. *Landscape Ecology* . 293-303.

Landau, B., L. B. Smith, et al. (1988). The importance of shape in early lexical learning. *Cognitive Development* **3**: 299-321.

Lee, D. R. and G. T. Sallee (1970). A Method of Measuring Shape. *Geographical Review.* 555-63.

Lo, C. P. (1980). Changes in the Shapes of Chinese Cities. *Professional Geographer* . 173-183.

Lord, E. A. and C. B. Wilson (1984). *The Mathematical Description of Shape and Form.* West Sussex, England, Ellis Horwood Limited.

MacEachren, A. M. (1985). Compactness of geographic shape: comparison and evaluation of measures. *Geografiska Annaler.* 53-67.

Marr, D. (1982). *Vision.* San Francisco, W. H. Freeman and Company.

Moellering, H. and J. N. Rayner (1982). The dual axis fourier shape analysis of closed cartographic forms. *The Cartographic Journal* **19**(1): 53-59.

Pavlidis, T. (1978). A Review of Algorithms for Shape Analysis. *Computer Graphics and Image Processing* . 243-258.

Preparata, F. P. and M. I. Shamos (1985). *Computational Geometry: An Introduction.* New York, Springer-Berlag.

Rhind, D., P. Armstrong, et al. (1988). The Domesday Machine: A nationwide Geographical Information System. *The Geographical Journal* **154**(1): 56-68.

Simons, P. (1974). Measuring Shape Distortions of Retail Market Areas. *Geographical Analysis* . 331-340.

Smith, A. H. (1966). Perception of shape as a function of order of angles of slant. *Perceptual and motor skills* **22**: 971-978.

Xia, L. (1996). A method to improve classification with shape information. *International Journal of Remote Sensing* **17**(8): 1473-1481.

LINEAR-TIME SLEEVE-FITTING POLYLINE SIMPLIFICATION ALGORITHMS

Zhiyuan Zhao, Alan Saalfeld
Department of Civil and Environmental Engineering and Geodetic Science
The Ohio State University, Columbus, OH 43210, USA
zhao.29@osu.edu, saalfeld.1@osu.edu

ABSTRACT

We present three variants of a polyline simplification algorithm. The basic algorithm uses a variable angle tolerance measure to find maximal subsequences of vertices of the polyline that may be replaced by a single segment in a simplified approximation. In our key theoretical development, we prove that an easily implemented angle-testing procedure is locally equivalent to ε-buffering; then we demonstrate that we may iterate the angle-testing procedure to find a maximum sleeve (rectangular strip in 2-D) of width 2ε that starts at any vertex $\mathbf{p_i}$ and contains successive vertices $\mathbf{p_{i+1}}, \ldots, \mathbf{p_{j-1}}, \mathbf{p_j}$. The sleeve is maximum in the sense that it is the rectangular strip of width 2ε that covers the largest number **(j-i+1)** of consecutive vertices starting with $\mathbf{p_i}$. We proceed to build the longest possible sleeve from $\mathbf{p_0}$ to some $\mathbf{p_i}$, then from $\mathbf{p_i}$ to some $\mathbf{p_j}$, and so on, until we have covered the entire polyline with "long sleeves". The center-line (or a near-center-line) of each sleeve offers a one-segment approximation to the sub-polyline of the original polyline linking of all of the consecutive vertices inside the sleeve. The three variants of our basic algorithm are the result of using different criteria to "trim the sleeve".

BACKGROUND

Our approach to polyline simplification applies unconstrained local processes to the polyline to produce a simplified polyline that lies within a given prescribed distance ε of the original polyline. We process the vertices of the polyline in order; and at any stage in our processing, all vertices are partitioned into three subsequences: the first **i+1** vertices $\{\mathbf{p_0}, \mathbf{p_1}, \ldots, \mathbf{p_i}\}$, that have already been completely processed (i.e., each vertex has received its final classification of "in" or "not in" the simplified polyline), the next **k** vertices $\{\mathbf{p_{i+1}}, \mathbf{p_{i+2}}, \ldots, \mathbf{p_{i+k}}\}$, that belong to an active pool of vertices currently undergoing processing, and the final **n-i-k** vertices $\{\mathbf{p_{i+k+1}}, \ldots, \mathbf{p_n}\}$, that have yet to be processed. Our selection algorithms are local because they only operate on the pool of vertices $\{\mathbf{p_{i+1}}, \mathbf{p_{i+2}}, \ldots, \mathbf{p_{i+k}}\}$, adding one candidate vertex $\mathbf{p_{i+k+1}}$, at a time on the right and removing a variable number of vertices $\mathbf{p_{i+1}}, \mathbf{p_{i+2}}, \ldots$, from the left hand side of the pool (as soon as they have been classified as "in" or "not in" the simplified line). Because the pool size **k** is not bounded, our local algorithm is called "unconstrained" (McMaster, 1992). Instead of using a distance tolerance directly as our nearness threshold, we convert each distance tolerance into a variable angle tolerance for testing the vertices of our polyline. Our sequential process places each successive polyline vertex into a pool of candidates for

possible deletion or inclusion. As the pool of candidates grows, the angle tolerance (angle range) in which we search to find a candidate vertex that will allow us to delete the entire pool grows smaller. When the angle range becomes empty, then we must include in our simplified polyline at least one vertex from the candidate pool. Our methods behave like a local filter that throws away unnecessary vertices and retains necessary vertices for the simplified polyline.

In this paper, we first present a greedy algorithm that deletes polyline vertices as they are found to lie within locally computed angle tolerances (and keeps them in the simplified polyline when they are not). The greedy-deletion algorithm is simple and fast (linear running time), but may fail to delete some clearly unnecessary vertices from the approximating polyline. To overcome this drawback, we designed a second algorithm that postpones decisions about certain ambiguously situated vertices in the candidate pool. The second algorithm removes more vertices than the first algorithm, but the polyline with fewer vertices still successfully approximates the original polyline. The second variant, however, requires more complex processing of the pool values; and the worst-case complexity of the second algorithm is no longer linear. We finally offer a third variant that allows vertices to be perturbed slightly to obtain even simpler polyline representations within a prespecified threshold. This relaxation of the constraint on location of vertices for the simplifying polyline not only further reduces the number of output vertices, but also actually recovers the linear time complexity of the first variant.

We analyzed our basic and modified algorithms mathematically and compared them to the Douglas-Peucker and other algorithms. We also tested our methods empirically on real cartographic data and on special short vector data produced by vectorizing raster images into Freeman-code vectors. We found our method especially suitable for large, dynamic data sets and real-time processing because we do not need to store all original data at one time; and we do not need to preprocess data before simplification. Our sequential local processes produce a satisfying overall appearance of the output. Downloadable C++ code, a more extensive write-up of empirical test results of our algorithms, and an interactive Java demonstration have been set up on a web page (Zhao, 1996b)

MATHEMATICAL PRELIMINARIES

Distances and angles are two related geometric measures for determining point selection or rejection in polyline simplification. We present a ***sector bound*** as another geometric measurement useful in polyline simplification. Geometrically, a sector bound is a swept angle emanating from a distinguished point. Mathematically, it may be used to constrain segments emanating from that distinguished point to pass within a threshold distance of all points in a set of points. A sector bound is represented by two angles. It is easily computed and updated by local processes. With sector bounds we build polyline simplification

algorithms that process a polyline's vertices in order (locally) and yet produce a guaranteed satisfying overall approximation of the polyline.

Suppose that **p** and **q** are two points on the plane, which we write $\mathbf{p}, \mathbf{q} \in \mathbf{R}^2$. We will use the following notation:

$\mathbf{L(p,q)}$ denotes the directed line segment from point **p** to point **q**. (This means $\mathbf{L(p,q)} \neq \mathbf{L(q,p)}$.)

$\alpha(\mathbf{p,q})$ signifies the angle measured counter-clockwise from the positive X-axis to the directed line corresponding to the line segment $\mathbf{L(p,q)}$.

$\mathbf{d(p,q)}$ will represent the distance from a point **p** to the point **q.**

We define $\mathbf{d(p_1,q,p_2)}$ to be the perpendicular distance from point **q** to the line passing through points $\mathbf{p_1}$ and $\mathbf{p_2}$ ($\mathbf{p_1}, \mathbf{p_2} \in \mathbf{R^2}$). Notice that we write the vertices in an unusual order. The reason for this unusual choice is that in our polyline vertex sequence, the vertices will actually appear in this order; and the intermediate vertex **q** will be tested for significant displacement from the approximating segment joining neighbor points $\mathbf{p_1}$ and $\mathbf{p_2}$. We also have the following definition: The *Sector Bound* (or *swept sector*) $\mathbf{A(p, \alpha_1, \alpha_2)}$ of point **p** and angles α_1 and α_2 is a point set given by:

$$\{\mathbf{q} \in \mathbf{R}^2 \mid \alpha_1 \leq \alpha(\mathbf{p}, \mathbf{q}) \leq \alpha_2\}$$

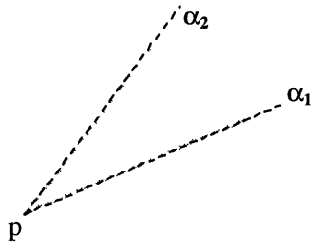


Figure 1. Sector bound

The sector bound $\mathbf{A(p, \alpha_1, \alpha_2)}$ of point **p** is described by the two angles: start angle α_1 and finish angle α_2 . Figure 1 shows a sector bound. We may always assume that α_1 is less than or equal to α_2, and the angle from α_1 to α_2 in the counter-clockwise direction is positive and less than 360°. For example, if α_1=350° and α_2=15°, then we express α_2 as 375° (=15°+360°) and α_1 as 350° to make the counter-clockwise difference positive to correspond to the usual order of the real numbers.

Lemma 2.1. The intersection of two sector bounds $\mathbf{A(p,\alpha_{11},\alpha_{12})}$ and $\mathbf{A(p,\alpha_{21},\alpha_{22})}$ which have the same initial point **p** is again a sector bound with same initial point **p**. $\mathbf{A(p, \alpha', \alpha'')} = \mathbf{A(p, \alpha_{11}, \alpha_{12})} \cap \mathbf{A(p, \alpha_{21}, \alpha_{22})}$, where $\alpha' = \boldsymbol{max}\{\alpha_{11},\alpha_{21}\}$, the larger of the start angles; and $\alpha'' = \boldsymbol{min}\{\alpha_{22},\alpha_{12}\}$, the smaller of the finish angles.

Proof: It is clear from the geometry in Figure 2. The intersection is $\mathbf{A}(\mathbf{p}, \alpha', \alpha'')$ with $\alpha' = \alpha_{21}$ and $\alpha'' = \alpha_{12}$. ■
If $\alpha' > \alpha''$, then we have that the intersection $\mathbf{A}(\mathbf{p},\alpha',\alpha'')$ is empty.

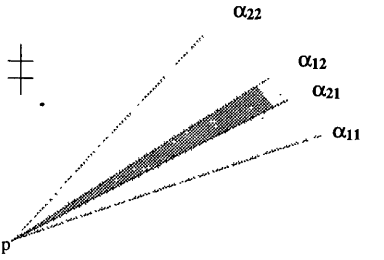


Figure 2. Intersection of sector bounds

Next we give the following definition: The *Epsilon Sector* $\mathbf{Q}(\mathbf{p}_1,\mathbf{q},\varepsilon)$ of point $\mathbf{q}$ from point $\mathbf{p}_1$ with threshold ε is a point set given by $\{\mathbf{p}_2 \in \mathbf{R}^2 | \mathbf{d}(\mathbf{p}_1,\mathbf{q},\mathbf{p}_2) \leq \varepsilon\}$.

Notice that $\mathbf{p}_2$ does not need to be near to $\mathbf{q}$. $\mathbf{p}_2$ only needs to determine a direction together with $\mathbf{p}_1$ that passes near to $\mathbf{q}$. A line through point $\mathbf{p}_1$ and a point $\mathbf{p}_2$ in $\mathbf{Q}(\mathbf{p}_1,\mathbf{q},\varepsilon)$ has the perpendicular distance from $\mathbf{q}$ no larger than ε. We can use the epsilon sector for polyline simplification. The segment from $\mathbf{p}_1$ to $\mathbf{p}_2$ is the simplified line segment and $\mathbf{q}$ is an original point. The point $\mathbf{q}$ has perpendicular distance $\leq \varepsilon$ to line $(\mathbf{p}_1, \mathbf{p}_2)$, hence point $\mathbf{q}$ can be deleted if $\mathbf{p}_1$ and $\mathbf{p}_2$ are selected. In our line simplification process, $\mathbf{p}_1$ is both the end point of the last accepted line segment in the sequential polyline building process and the starting point of the next simplified line segment that will be added to the already processed initial sequence of points. The three points are consecutive points of the original line. We can delete the point $\mathbf{q}$ if and only if the point $\mathbf{p}_2$ is in $\mathbf{Q}(\mathbf{p}_1,\mathbf{q},\varepsilon)$. Next, we will examine the relationship between the epsilon sector and the sector bound.

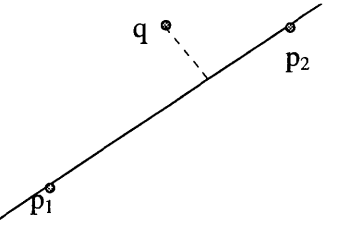


Figure 3. Epsilon sector

Theorem 2.1: When $\mathbf{d}(\mathbf{p},\mathbf{q}) \geq \varepsilon$, the Epsilon Sector $\mathbf{Q}(\mathbf{p},\mathbf{q},\varepsilon)$ is equivalent to the sector bound $\mathbf{A}(\mathbf{p}, \alpha_1, \alpha_2)$ with

$$\alpha_1 = \alpha(\mathbf{p},\mathbf{q}) - \delta,\ \alpha_2 = \alpha(\mathbf{p},\mathbf{q}) + \delta, \text{ where } \delta = \mathbf{sin}^{-1}(\varepsilon/\mathbf{d}(\mathbf{p},\mathbf{q})).$$

Proof: Suppose $\mathbf{v} \in \mathbf{Q}(\mathbf{p},\mathbf{q},\varepsilon)$. Then according to the definition of epsilon sector, we have $\mathbf{d}(\mathbf{p},\mathbf{q},\mathbf{v}) \leq \varepsilon$. Denote $\phi = \| \alpha(\mathbf{p},\mathbf{q}) - \alpha(\mathbf{p},\mathbf{v}) \|$, since $\mathbf{sin}\phi = \mathbf{d}(\mathbf{p},\mathbf{q},\mathbf{v})/\mathbf{d}(\mathbf{p},\mathbf{q})$, $\mathbf{sin}\delta = \varepsilon/\mathbf{d}(\mathbf{p},\mathbf{q})$, and $\phi, \delta \in [0°, 90°]$, we get $\phi \leq \delta$. That is: $\alpha(\mathbf{p},\mathbf{q}) - \delta \leq \alpha(\mathbf{p},\mathbf{v}) \leq \alpha(\mathbf{p},\mathbf{q}) + \delta$. According to the definition of sector bound, $\mathbf{v} \in \mathbf{A}(\mathbf{p},\alpha_1,\alpha_2)$, where $\alpha_1 = \alpha(\mathbf{p},\mathbf{q}) - \delta$, $\alpha_2 = \alpha(\mathbf{p},\mathbf{q}) + \delta$, $\delta = \mathbf{sin}^{-1}(\varepsilon/\mathbf{d}(\mathbf{p},\mathbf{q}))$.

Conversely, if $\mathbf{v} \in \mathbf{A}(\mathbf{p},\alpha_1,\alpha_2)$ with $\alpha_1 = \alpha(\mathbf{p},\mathbf{q}) - \delta$, $\alpha_2 = \alpha(\mathbf{p},\mathbf{q}) + \delta$, where $\delta = \mathbf{sin}^{-1}(\varepsilon/\mathbf{d}(\mathbf{p},\mathbf{q}))$, then $\alpha(\mathbf{p},\mathbf{q}) - \delta \leq \alpha(\mathbf{p},\mathbf{v}) \leq \alpha(\mathbf{p},\mathbf{q}) + \delta$. Thus, we see that $\phi = \| \alpha(\mathbf{p},\mathbf{q}) - \alpha(\mathbf{p},\mathbf{v}) \| \leq \delta$, since $\mathbf{sin}\phi = \mathbf{d}(\mathbf{p},\mathbf{q},\mathbf{v})/\mathbf{d}(\mathbf{p},\mathbf{q})$, $\mathbf{sin}\delta = \varepsilon/\mathbf{d}(\mathbf{p},\mathbf{q})$, and $\phi, \delta \in [0°, 90°]$, we get $\mathbf{d}(\mathbf{p},\mathbf{q},\mathbf{v}) \leq \varepsilon$. According to the definition of epsilon sector, we have $\mathbf{v} \in \mathbf{Q}(\mathbf{p},\mathbf{q},\varepsilon)$. ■

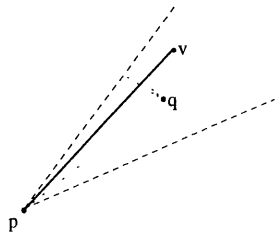


Figure 4. Sector bound and epsilon sector

This theorem tells us that an epsilon sector is geometrically equivalent to a sector bound. Since a sector bound is far easier to calculate and maintain, we will use the sector bound for polyline simplification to find new points of the simplified line segments. This means we may convert a global problem of epsilon sector determination into a local process of determining sector bounds. To determine if a point **q** is inside a given band of tolerance ε of a line segment from point **p** to point **v**, we determine if a point **v** is inside the sector bound $\mathbf{A}(\mathbf{p},\alpha_1,\alpha_2)$ which only necessitates calculation of the angle $\alpha(\mathbf{p},\mathbf{v})$.

Let consider a polyline, a sequence of line segments. Here we will use the symbol "$\cap$" (intersection) to represent the intersection set of epsilon sectors.

Theorem 2.2: Suppose a polyline has vertices $\{\mathbf{p_i} \mid \mathbf{i=0,1,\ldots k}\}$. Then there exists a point **q** such that all points $\mathbf{p_i}$ **(i=1, ... k)** have perpendicular distance to line $\mathbf{L(p_0,q)}$ within a given tolerance ε if and only if

$$\cap \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon) \neq \varnothing.$$

Proof: If such a point **q** exists, then all points $\mathbf{p_i}$ **(i=1, ..., k)** have perpendicular distance to line $\mathbf{L(p_0,q)}$ less than the tolerance ε, $\mathbf{d(p_0, p_i, q)} \leq \varepsilon$ **(i=1, ... k)**, that is $\mathbf{q} \in \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon)$ **(i=1, ... k)**. So $\cap \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon) \neq \varnothing$.

If $\cap \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon) \neq \phi$, suppose $\mathbf{q} \in \cap \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon) \neq \phi$, then $\mathbf{q} \in \mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon)$ **(i=1,... k)**, that is $\mathbf{d(p_0, p_i, q)} \leq \varepsilon$ **(i=1, ... k)**, So all points $\mathbf{p_i}$ **(i=1, ... k)** have perpendicular distance to line $\mathbf{L(p_0,q)}$ within the given tolerance ε. ■

Corollary 2.1: Suppose a polyline has vertices $\{\mathbf{p_i} \mid \mathbf{i=0,1,\ldots k}\}$. Then there exists a point **q** such that all points $\mathbf{p_i}$ **(i=1, ... k)** have perpendicular distance to line $\mathbf{L(p_0,q)}$ within a given tolerance ε if and only if

$\alpha' \leq \alpha''$, where $\alpha' = \mathit{max}\ \{\ \alpha_{1i} \mid \mathbf{i=1,...,k}\ \}$, and $\alpha'' = \mathit{min}\ \{\ \alpha_{2i} \mid \mathbf{i=1,...,k}\ \}$.

Here α_{1i} and α_{2i} are the two angles of sector bounds $\mathbf{A}(\mathbf{p},\alpha_{1i},\alpha_{2i})$ equivalent to $\mathbf{Q}(\mathbf{p_0}, \mathbf{p_i}, \varepsilon)$ **(i=1, ... k)**.

Proof: Combine Lemma 2.1 and Theorem 2.1 to get the conclusion. ■

This theorem obviously gives us a new opportunity for line simplification. For a polyline, the sector bound intersection is the only possible location for a subsequent point that can simplify the pending chain of points with a single segment. The second point of the simplified segment must lie inside this intersection. If this intersection is empty, there will exist no simplified segment that meets the distance tolerance condition. Figure 5 illustrates such a polyline.

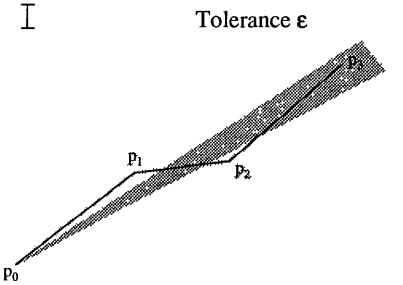


Figure 5. Epsilon bounds intersection of a polyline

In Figure 5, we have a polyline of points $\{\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}\}$. The simplified segment will start from point $\mathbf{p_0}$. For the given distance tolerance ε, the left-upper sector (shown as a triangle) is the epsilon bound $\mathbf{Q}(\mathbf{p_0},\mathbf{p_1},\varepsilon)$; the right-down sector (also shown as a triangle) is the epsilon bound $\mathbf{Q}(\mathbf{p_0},\mathbf{p_2},\varepsilon)$. They have a intersection shown as a darker area. Since this intersection is not empty, we can delete points $\mathbf{p_1}$ and $\mathbf{p_2}$ and use a point in the intersection to form a new line segment to approximate the original lines. We see point $\mathbf{p_3}$ is in the intersection, so line $\mathbf{p_0}$ to $\mathbf{p_3}$ is the simplified line of the original polyline.

Theorem 2.1 tells us that the sector bound and the epsilon sector are geometrically equivalent. We will see, however, that it is much easier to work with sector bounds than with epsilon sectors.

We are now ready to describe and examine our three algorithm variants.

A GEOMETRIC DESCRIPTION OF OUR ALGORITHMS

Imagine trying to fit a sleeve of width 2ε to the first $\mathbf{k}$ points in our polyline. If $\mathbf{k=2}$, then this is easy. Suppose that the sleeve fits the first $\mathbf{k}$ points; and we want to adjust it (if necessary) to fit the $(\mathbf{k+1})^{st}$ point as well. It is clear that there will only be a limited amount of "play" in the sleeve to realign it. It is also clear that if we keep the first point $\mathbf{p_0}$ in the center of the sleeve that the "play" in the sleeve will correspond to a swept angle, our sector bound above. Our mathematical preliminaries have guaranteed that we can slide the sleeve forward as far as possible with a very fast and efficient sector bound update computation.

The missing step in our polyline simplification routine handles

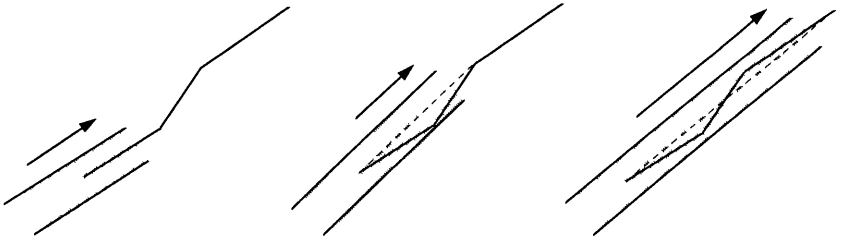


Figure 6. A sleeve moves along the polyline covering consecutive vertices.

the case for which the next point $\mathbf{p_k}$ cannot be included in the sleeve. We must advance and reset the starting point for our sleeve algorithm and also decide what to do with the points that had been intermediate vertices inside the sleeve. All of the points inside the sleeve can certainly be approximated by the sleeve centerline to within ε; and choosing the sleeve centerline as a segment in an approximating simplified polyline is one of the variants that performs well. If we choose the centerline, then we may throw away all of the intermediate points inside the sleeve. Choosing the centerline end points may force us to choose an approximating vertex that is not one of the original polyline vertices. Nonetheless the centerline end point will be within ε of the final $\mathbf{k}^{th}$ vertex $\mathbf{p_{k-1}}$.

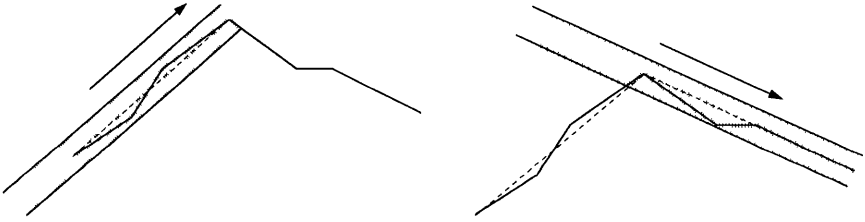


Figure 7. When a vertex cannot fit in the sleeve, a new sleeve is begun

If we simply choose the last vertex $\mathbf{p_{k-1}}$ as the approximating segment end point, then we will only guarantee that our approximating segment is within 2ε of the original polyline. We may turn this logic around and use it to our advantage by building a sleeve of width ε instead of 2ε. In that case, we may simply accept the final vertex that fits in the sleeve as our current approximating segment's end point and our next approximating segment's starting point. If we employ the straightforward narrower ε-sleeve strategy, however, we may wind up choosing many more vertices than necessary for our simplifying polyline.

A middle-of-the-road option that uses a 2ε sleeve, but only subsamples vertices of the original polyline (i.e., does not create new approximating vertices), requires a special subroutine to handle vertices in the sleeve after the sleeve's vertex set has reached a limit. The special subroutine will advance the starting vertex and decide whether to keep intermediate vertices as vertices of the simplifying polyline. There are several options for this special subroutine; and

no single option appears to guarantee the fewest number of vertices in the approximating polyline for all input polylines. If the sleeve contains vertices $\{p_{i+1}, p_{i+2}, \ldots, p_{i+k}\}$, where p_i is the current starting vertex, then we choose the vertex p_{i+j} with the largest subindex in the sleeve such that $L(p_i,p_{i+j})$ approximates the subpolyline chain $p_i, p_{i+1}, p_{i+2}, \ldots, p_{i+j}$ to within ε. Then we make p_{i+j} the new starting vertex. There will always be such an approximating vertex, although in the worst case that vertex might be p_{i+1}. Even though the new sleeve from the new starting vertex p_{i+j} will clearly contain the subsequent vertices $p_{i+j+1}, p_{i+j+2}, \ldots, p_{i+k}$ that had been in the old sleeve, we must, nevertheless, recompute all of the sector bounds since the starting vertex has changed, hence the swept angle is different in every case. This requirement to recompute the sector bounds can become expensive; and it may render our algorithm complexity quadratic instead of linear as it had been earlier.

GENERAL ALGORITHM DESCRIPTION AND PSEUDOCODE

We summarize the common components of our three algorithms below:

Initialization:

```
(1)   Set starting vertex to p0;
(2)   Set vertex set of simplifying line to {p0};
(3)   Set sleeve set to ∅;
(4)   Set sector bound to full 360° range: ([0°,360°]).
```

Iterative introduction of vertices $\{p_1, p_2, \ldots, p_n\}$ to processing:

```
(5)    For i = 1 to n {
(6)      While pi is not in the current sector bound {
(7)        Update the starting vertex to some pj* in the sleeve set;
(8)        Update the sleeve set by removing all points through pj;
(9)        Update the vertex set of simplifying line by adding pj*;
(10)       Recompute sector bound for new start vertex, sleeve set;
(11)     }
(12)     Add pi to the sleeve set;
(13)     Recompute the sector bound;
(14)   }                                        // End of For loop.
(15)   Add pn* to the vertex set of the simplifying line.
```

The variants to the basic algorithm given above require further specification of the three update statements (7) through (9) and statements (10) and (15). Here are the details:

a) For the algorithm that uses the narrower sleeve (of width ε), the value of p_j^* in steps (7) and (9) is p_{i-1}; and the value of p_j in step (8) is also p_{i-1}. The sector bound for the empty sleeve set in (10) is once again always the full 360° range. The value for p_n^* in step (15) is p_n.

b) For the algorithm that uses the wider sleeve (of width 2ε) and allows perturbation of vertices, the value of p_j^* in steps (7) and (9) is the point on the sleeve centerline that is closest to p_{i-1}; and the value of p_j in step (8) is

the original polyline's vertex $\mathbf{p_{i-1}}$. The sector bound for the empty sleeve set in (10) is once again always the full 360° range. The value for $\mathbf{p_n}^*$ in step (15) is the point on the sleeve centerline that is closest to $\mathbf{p_n}$.

c) For our middle-of-the-road variant with a 2ε width sleeve, the update procedures, including recomputing the sector bound in step (10), may require several steps. As we are adding vertices to the growing sleeve, we may easily keep track of those vertices that are capable of forming, along with the current starting point, a single line segment that adequately approximates all of the intermediate vertices between the currently added vertex and the current starting vertex. A vertex will provide the best kind of single segment approximator if and only if that vertex actually falls within the current sector bound. We will choose $\mathbf{p_j}^*$ in steps (7) and (9) and $\mathbf{p_j}$ in steps (8) to be the one such simplifying vertex $\mathbf{p_j}$ with the largest index $\mathbf{j \leq i-1}$.

In both variants a) and b), we clearly have linear time performance because each sleeve is augmented one vertex at a time until it cannot be augmented further. At that point, the entire subsequence of vertices within the sleeve is "retired"; and a new sleeve is begun from where the last one ended. There is no backtracking; and each vertex of the original polyline appears once in exactly one sleeve building operation.

CONCLUSION

We showed how to use a sector bound calculation to assign maximum consecutive sequences of polyline vertices to buffer "sleeves". Because the sequences are as large as possible subject to constraints on approximation threshold settings, we produce a rather reduced number of segments in our simplifying polylines. We have conducted empirical tests to compare the performance of the three variants to each other and to the classic Douglas-Peucker algorithm for polyline simplification. The results of those tests were very favorable for our techniques, both in appearance and in quantitative measurements of vertices used in the simplification. The results of those experiments and more information on the algorithms themselves, including complete working code, are available to the interested reader at the World Wide Web site ***http://ra.cfm.ohio-state.edu/grad/zhao/algorithms/linesimp.html***.

Our key practical result is the use of the sector bound as a new measurement of line simplification; and our key theoretical result is the proof that the sector bound, an easily maintained measurement, is locally geometrically equivalent to an ε-buffer strip of the type used in the classic Douglas-Peucker algorithm.

Finally we mention that future work is suggested by our choice of the descriptor "sleeve" and its extended meaning in 3-D. Our "sector bound" in 3-D is not just a single cone, but an intersection of cones. A sleeve, however, is nothing more nor less than a right circular cylinder having radius ε.

ACKNOWLEDGEMENTS

The authors would like to thank Paula Stevenson for providing test data. Thanks also go to Dr. Raul Ramirez and Dr. John Bossler of the Ohio State University Center for Mapping for their support in this research.

REFERENCES

Chrisman, Nicholas R., 1983, Epsilon Filtering: A Technique for Automated Scale Changing, ***Technical Paper of 43rd Annual ACSM Meetings***, Washington D.C., 322-331.

Cromley, Robert G., 1991, Hierarchical Methods of Line Simplification, ***Cartography and Geographic Information System***, 18(2):125-131.

Douglas, David H. and Thomas K. Peucker, 1973, Algorithms for Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature, ***The Canadian Cartographer***, 10(2):112-123.

Imai, Hiroshi, and Masao Iri, 1986, Computational Geometric Methods for Polygonal Approximations of a Curve, ***Computer Vision, Graphics, and Image Processing***, 36:31-41.

McMaster, Robert B., and K. Stuart Shea, 1992, ***Generalization in Digital Cartography***, Washington, DC: Association of American Geographers.

Williams, C.M., 1978, An Efficient Algorithm for the Piecewise Linear Approximation of a Polygonal Curves in the Plane, ***Computer Vision, Graphics, and Image Processing***, 8:286-293.

Zhao, Z., 1996, Java Gallery of Geometry Algorithm, ***http://ra.cfm.ohio-state.edu/grad/zhao/algorithms/linesimp.html***

Zhao, Z. and Alan Saalfeld, and Raul Ramirez, 1996, A General Line-Following Algorithm for Raster Maps, ***Proc. of GIS/LIS'96 Conference***, Denver, CO, 85-93.

PUBLIC PARTICIPATION GEOGRAPHIC INFORMATION SYSTEMS

Timothy Nyerges, Department of Geography
University of Washington, Seattle, WA 98195, USA
nyerges@u.washington.edu

Michael Barndt, Department of Urban Studies
University of Wisconsin - Milwaukee, Milwaukee, WI, 53201, USA
mbarndt@csd.uwm.edu

Kerry Brooks, Department of Planning and Landscape Architecture
Clemson University, Clemson, SC 29634, USA
kerry@vito.arch.clemson.edu,

ABSTRACT

Increasing societal inclination towards participatory democracy is encouraging research on the development and use of public participation geographic information systems (PPGIS). This paper summarizes three scenarios describing actual and/or potential use of a PPGIS. One scenario concerns a public-private coalition strategy for brownfield development (urban land rehabilitation) in and near Seattle, WA. A second scenario concerns neighborhood crime watch in Milwaukee, WI. A third scenario concerns forest conservation planning and action in the Southern Appalachian Mountains. Generalizations are drawn from these scenarios to synthesize a table of general requirements for a PPGIS. Concluding comments assess the current state of development and address future prospects.

Keyword: GIS-participatory, GIS-society, collaboration, public-participation

1. INTRODUCTION

A societal trend toward shared decision making about public concerns is a basic motivating factor that encourages research on the development and use of public participation geographic information systems (PPGIS). Some of these public concerns include decisions about resources and environment that involve land use planning (Duffy, Roseland, Gunton 1996), strategies for planning a citizen crime watch, and plan development for forest conservation and sustainable use of limited natural resources (Diamond and Noonan 1996). The primary rationale for enhanced public participation in the decision process is based on the democratic maxim that those affected by a decision should participate directly in the decision making process (Smith 1982).

Current GIS technology has been developed mainly to support organizational use of GIS (Campbell and Masser 1995). Such developments can be labeled first generation GIS, or GIS/1. Because of the need to expand the access to geographic information for participatory kinds of activity, as well as more personal use of geographic information , many researchers are recognizing the need for a new kind of GIS which has been called GIS/2, or second generation GIS (http://ncgia.spatial.maine.edu/ppgis/criteria.html). Personal GIS and PPGIS compose GIS/2.

The goal of this paper is to articulate a set of functional requirements for PPGIS. Sandman (1993) has identified nine publics relevant to discussion about community problems. The publics are: neighbors, concerned citizens, technical experts, media, activists, elected officials, business and industry, and local, state and federal government regulators. Which of these publics is involved in any particular scenario of PPGIS use depends on several characteristics, for example, topic of concern, geographic location, meeting venue, public process, and technology used to facilitate conversation. In this paper we present three application scenarios for use of a PPGIS. From these three scenarios we synthesize a sense of the overall system requirements for a generic PPGIS.

2. APPLICATIONS OF PUBLIC PARTICIPATION GIS

Three application scenarios described below were selected so as to capture a breadth of issues about public process and how a PPGIS might be used.

2.1 Public Participation in Brownfield Development Usi ıg GIS

Brownfield projects are public-private partnerships for urban industrial land parcel rehabilitation. A general strategy for a brownfield project is to clean a land parcel(s) to a level which puts the land back into productive use. However, clean-up standards in local jurisdictions must meet federal and state across-the-board-standards that commonly are more stringent than needed for the land use activity actually practiced in those areas. Consequently, land costs are prohibitively high because clean-up costs become part of the transaction. A major challenge is therefore to have commercial/industrial, regulatory and financial organizations collaborate to achieve a "brownfield" level of clean rather than "greenfield" level of clean.

The Duwamish Coalition in Seattle, Washington is responsible for one of seventy-six brownfield pilot projects co-sponsored by the U. S. EPA headquarters or regional offices (Institute for Responsible Management 1996). Convened in April 1994, the Coalition is composed of representatives from small and large businesses; labor unions; local, state and federal governments; Native American Tribes; environmental and community organizations; and local banks, business associations and educational institutions. The Duwamish Corridor, includes a mix of industrial, commercial and residential land uses, marine terminals and transportation infrastructure, open space/parks, public

facilities, landfill sites, Superfund sites, habitat restoration areas, and tribal fishing areas along the Duwamish Waterway. The mission of the Duwamish Coalition (Duwamish Coalition 1996) is to "...preserve and reclaim industrial land for the purposes of expanding the manufacturing and industrial job base, and protecting and enhancing the natural environment of the corridor. ... Meetings of the Coalition's Steering Committee and Task Forces are open to the public", with public interpreted to mean all who want to participate.

2.1.1 Current Scenario. The Coalition's brownfield development activity is incremental in nature, and part of a policy implementation strategy for urban growth management which focuses on land use densification in mixed-land use areas. To plan the development activity, four types of group meetings, hence clusters of participants, carry out the work: staff meetings, task force meetings, steering committee meetings, and annual summit meetings.

For the most part, standard communication technology has been used to support communication in face-to-face group meetings, these technologies being slide projectors, transparencies, posters, and hardcopy print. In some face-to-face meetings the Coalition made use of hardcopy maps based on geographic data from City of Seattle and King County Arc/Info data files. Poster size maps were available for review at meetings, and notebook-sized (8.5" x 11") maps were made for review at other times. In other words, standard access to paper-based documents was the only mode supported for access to geographic data.

To help expand the distribution of Coalition information, a WWW site was developed that includes their mission statement and brief description of activity (Duwamish Coalition 1996). Due to limited resource availability little of the details of the Coalition discussion appear on the WWW site, including the GIS maps reviewed in various meetings that provided information overview.

2.1.2 PPGIS Needs Revealed. We assume here that topic, place, and publics participating will likely stay the same for the brownfield collaborative effort, but venue and technology can change the nature of the participation process, and thus reveal a new set of needs for enhancing participation. The venue is affected mostly by available communication technology. With changes in communication process, the other three technologies, data management, computer map graphics, and spatial analysis, are likely to be influenced as well.

In regards to communication management, dialogue is constrained by meeting venue. Although everyone is invited, few have the time to attend all face-to-face meetings. The frequency of meetings and the extended process required to carry through on a topic hamper participation. Together, those constraints limit a group's ability to undertake information synthesis.

In regards to information management, a lack of resources to make GIS data available is a major barrier to a balanced dialogue among the publics. Wide distribution of all geographic information among participating publics should be a goal. Satisfactory public review of the data used to generate the information takes time. GIS data management techniques open to all publics at

any time/library can enhance participation, including an ability to collect their own data in support of their arguments.

In regards to map display, several members of the Coalition often discussed patterns on maps in the various meetings. Group memory of their discussions can be enhanced by providing text hot links to descriptions of the topics discussed. Other maps were used to assess the priorities of areas to be developed, each stakeholder group having their own priorities. New types of maps that depict both the individual stakeholder group as well as overall consensus priorities among the stakeholder groups could be useful.

In regards to spatial analysis, although King County staff created several maps, many groups would like to have access to GIS software and data to undertake their own GIS analysis. GIS can be used to assist with risk assessment at three levels: vulnerability analysis, screening analysis, and probability analysis. For vulnerability analysis, all potential hazards and receptors are identified in a broad-based approach to estimating potential risk. For screening, the most significant hazards and receptors are identified for which a risk estimate is provided. For probability analysis, a probability range is computed for biomarkers relevant to receptors based on a screening analysis.

2.1.3 Discussion. The four types of technologies mentioned above -- communication, database management, computer graphics display, and spatial analysis -- are fundamental for opening brownfield discussions to a wider audience. All meeting venues are important in facilitating the participation process. Unfortunately, same-time, same-place meetings are the only ones supported, whereas different-time and/or different-place meetings are not.

2.2 Public Participation in Urban Crime Surveillance

A second scenario concerns neighborhood residents discussing issues and options for a local response to urban crime issues. The Sherman Park community is an integrated community of black and white, middle and lower income residents. The community has been organized as an Association for more than 20 years to address issues in the neighborhood. The Crime Watch program has been in place for only five years, involving a volunteer pool of 60 volunteers and two staff members.

2.2.1 Current Scenario. Monthly meetings of the Crime Watch task force thrive on information. Calls to the Association, a printout of weekly police calls, and observations of the volunteer patrols through the neighborhood help in the review of problems the group must anticipate. Because GIS is not used at the current time, we speculate on how it might be used in the next subsection.

2.2.2 PGIS Needs Revealed. Information and visualization are required in real time. Imagine a computer with a large screen at the conference table. An Internet connection links data from the Police Department and the Metro Drug

Unit. The local network taps reports from Crime Watch patrols and calls from the neighborhood. General neighborhood demographics are also available.

The mapping system has been customized for the work of the task force. A parcel-based map of the neighborhood is the default screen. Symbols for each of the major categories of crime may be presented by selecting from a set of check boxes to the right side of the screen. Other attributes may also be selected to define the characteristics of parcels or other areas on the map. The parcel map may be generalized to block level patterns and statistical aggregates through a variety of choices on the left. At the bottom of the screen, a history bar allows selecting time periods for display - even animation.

Have burglaries on the east side of the community been increasing recently? Is this due in part to the intense anti-crime effort in the Metcalfe Park neighborhood further east? Adjust the frame to show the area of interest. Check the burglaries box to display. Widen the history bar to show six months of data at a time. Move the history bar back three years and start a slow speed animation. The trends are apparent on the screen. Select two sets of blocks within this area and the data may also be summarized as a time series bar graph.

Is a rash of assaults in one block related to reports of a new drug house in the community? Zoom into a four block area. Select assaults and the parcel attribute identifying reported drug houses. Set the history bar for a static display of the last three months. Save or print the results and reset the history bar to the same period one year ago.

Where have calls to the agency regarding crime been coming from? Select the "Calls" database. Leave all categories active. Set the history bar for the past month. A cluster of calls is apparent in one block. Is this an increase in crime or a more active block club? The task force would need to sort that out. Select one of the symbols on the map and notes about the call can be read.

2.2.3 Discussion. In these crime watch scenarios, comprehensive sources of current information are critical. The political process this represents will be more difficult than the technical issues are. Information also needs to be treated with caution. Some data includes confidential material. Access must be managed by a sophisticated database program aware of the level of information that can be made accessible to specific users. Information may often be incomplete. For example, a suspected drug house is recorded quite differently than an established drug house - if at all. Certain crimes may be unreported. Additional factors - the involvement of youth or gangs, relationship to drugs or the extent that a crime of assault involved persons within a family - may not be apparent in the data systems.

Neither data nor software can insulate the task force from the misuse of data, especially when a correlation seems obvious. A cause and effect conclusion can be influenced by the personal opinions of lay persons on the task force. Despite these limitations, residents provide a valuable perspective on the data from their experiences. Encouraging participatory review of data will add value that the professional analyst limited by formal data sets cannot achieve.

2.3 Public Participation in Forest Conservation Planning and Action

In this scenario, GIS is used for forest conservation planning by a collaborative of conservation groups. Led by the Chattooga River Watershed Coalition (CRWC), the planning team also included the Southern Appalachian Forest Coalition (SAFC) and the SE Regional Office of the Conservation Fund (CF). The geographic focus of plan is the Chattooga River watershed. Highlighted by the Chattooga River Wild and Scenic Corridor, the watershed is a globally significant hotspot for biological diversity and white-water recreation. The watershed contains about 70% public lands, and is trisected by the borders of three states and National Forests. Though CRWC was the key player, the participation of SAFC and CF was vital. These organizations provided staff and support, making the project possible. Additionally, GIS facilities were available in the Clemson University Dept. of Planning and Landscape Architecture, and a grant to CRWC funded a GIS-astute graduate student during Summer, 1995.

2.3.1 Current Scenario. A primary use of the project plan is as a citizens' alternative in the planning processes of the three National Forests. The collaborative evolved project requirements that included: (1) perform the GIS analysis on the CRWC-owned PC; (2) document all analyses, so that the process is repeatable by other conservation groups, and so that all aspects are open to scrutiny. (The latter involved a peer review process); (3) produce a *poster* and a *slick* document suitable for distribution as *public relations* pieces.

Since data limitations precluded calculating and overlaying various habitats as a suitability model would require, the collaborative adopted a strategy to further expand and protect (buffer) existing protected areas on a sub-watershed basis. The procedures applied in GIS demonstrate concepts developed by landscape ecologists (Forman and Godron 1986).

The group evaluated versions of the plan by comparing plan boundaries to known locations of resources. The final draft of August 1995 encompassed significantly more areas of importance within sensible boundaries. The final methodology was clear, defensible, and supported the vision of the participants.

Ten to fourteen persons attended meetings at critical points in the project. Other meetings were convened for reviewers and Board members. The venue for these meetings was *face-to-face (same time/same-place)*, although analyses planned at these meetings were performed *same-place/different-time*. The purpose of the meetings was to review or set direction and criteria for the GIS-based plan development. They depended on hard copy draft maps of data elements and analyses, usually produced with Arcview version 2.1. Attempts to use Arcview in real-time were not particularly successful due to lack of a projection system, and slow redraw times. Thus, letter sized maps, along with printing these same maps onto transparencies and projecting them, became the major geographic mode of communication. We documented the meetings and analysis design manually, on paper and white-boards.

The final phase of the project centered on peer reviews, and on producing the poster and document. These tasks took place in the geographically removed

offices of CRWC, SAFC, CF and the university. The graphic design firm was near SAFC, but distant from all others. Data, documents, products and revisions of necessity moved amongst all these locales -- the venue had shifted to *different-place/different-time*, and lacked effective supporting technology.

2.3.2 PPGIS Needs Revealed. A difficult task in this analysis was coping with the reality of the available data within the short time frame. Although a USFS project developed a GIS database, it was incomplete. Additionally, coalition-performed field work products needed digitizing and documentation. Data transfer amongst the venues and computers consumed scarce resources. Similarly, limitations of the PC and PC Arc/Info created additional work.

Lack of adequate (GIS) personnel and technological resources caused some problems. Without special support, this project would not have been possible using a GIS/1. It relied upon GIS expertise available *outside* the conservation groups. In particular, it depended upon work by knowledgeable graduate students. Losing a graduate student to another opportunity stymied GIS work late in the stages at the CRWC office. Tools that check for processing errors and which thoroughly document the processing undertaken would be helpful. Access to *smarter* GIS tools could make it possible to do without outside support, and make needed analyses more compatible with available resources. Finally, improved *what-if* tools flexible amongst venues would allow more robust and time-efficient collaboration in plan development.

A poster publication eventually took on a life of its own. Because the collaborative wanted the poster to have appeal beyond that of a *mere* map, it hired a graphic design firm. Identifying the firm, design, editing, communication, and data transfer difficulties added nearly one year to the process. A substitute for this process would facilitate local creation of *slick* output in any desired media.

The limitations encountered in the Chattooga process are summarized below. For communication management these include the need for tools which aid geographic communication across all meeting venues. In addition, there is a need for tools which interface seamlessly across output options and media (e.g., web, printing, CD-ROM).

For information management the tools include the need: to develop feasible ways to share data and results in a timely and inexpensive manner, including assurance of adequate access to the Internet or viable alternatives; to support field development of local data; to support critique of data and results; and the need to provide metadata tools which track and know the *meaning* of data bases, so that information is not lost due to staff and constituency changes within organizations. In addition there is a need for tools which record analyses processes such that the record of the analysis becomes part of the product, and at the same time a *model* usable by others.

For spatial analysis there is a need for tools with capabilities to substitute for human GIS expertise. There is a need for accessible tools to model population viability, and out-of-region externalities (e.g. air quality, continental

rarity). Given that groups successfully develop multiple sub-regional plans, we need tools to aid in their combination into a bio-regional plan.
2.3.3 Discussion. It is apparent that the needs of conservation groups encompass many venues, ranging from in-house personal analysis and in-house group collaboration, to interfaces with the planning and analysis processes of other agencies and groups in more public venues. The ideal PPGIS should support the full continuum of venues and modes of interaction if participation is to be fully encouraged.

3. CAPABILITIES IN A PUBLIC PARTICIPATION GIS

Generalizing across the scenarios we find that a GIS-enabled public participation process involves three phases: explore data to clarify issues (availability of data), establish a set objectives from what is known, and evaluate options about what is known. The three phases can each, more or less, make use of capabilities at two levels of sophistication (See Table 1). The two levels of sophistication are essentially "building block" levels, i.e., level 2 would not work effectively without level 1, but level 1 could stand alone.

Table 1. Functional Capabilities in a Public Participation GIS (adapted from Nyerges 1995)

Level 1: Basic information handling support
(a) Group Communication: idea generation and collection includes anonymous input of ideas, pooling and display of textual ideas, and search facilities to identify common ideas, (e.g., data/voice transmission, electronic voting, electronic white boards, computer conferencing, and large-screen displays)
(b) Information Management: storage, retrieval and organization of data and information (e.g., spatial and attribute database management systems)
(c) Graphic Display: visualization techniques for a specific part of a geographical problem (e.g., shared displays of charts, tables, maps, diagrams, matrix and/or other representational formats)
(d) Spatial Analysis: basic analysis functions (e.g. overlay and buffering)

Level 2: Enhanced analysis/discussion support
(e) Process Models: computational models that describe/predict the character of real-world processes (e.g., simulation models for describing changes in crime events or surface water flow across time);
(f) Decision Models: integration of individual criteria across aspects or alternatives, (e.g., multi-criteria decision models using multi-attribute and multi-alternatives for weighting rankings or preferences).
(g) Structured Group Process: methods for facilitating/structuring group interaction, (e.g., automated Delphi, nominal group techniques, electronic brainstorming, and technology of participation).

4. CONCLUSIONS AND FUTURE PROSPECTS

Given the interest in participatory decision making, there is a clear interest in PPGIS. The above needs and system requirements identified from these needs are coincident with many of the issues discussed and outlined in recent initiatives of the National Center for Geographic Information and Analysis. Particularly relevant is work performed under Initiative 17, Collaborative GIS (Densham, Armstrong and Kemp 1995) and Initiative 19, GIS and Society (Harris and Weiner 1996). In addition, a meeting on PPGIS held in summer, 1996, at the University of Maine refined these discussions (see http://ncgia.spatial.maine.edu/ppgis/ppgishom.html).

Some implementation progress is evident in these areas. Currently two flavors of developing PPGIS seem to exist. These reflect the degree of dependence upon skilled human operators, and the venues to with which they best fit. One flavor uses the expertise of a GIS analyst in the same-time/same-place venue to aid in group-based information exploration as described by Shiffer (1992) and Couclelis and Monmonier (1995), and group decision making as described by Nyerges (1995) and Jankowski et al. (1997). We term this flavor *soft-PPGIS*. In *soft-PPGIS*, the human chauffeur encapsulates needed system knowledge. The GIS support from a technically knowledgeable person in a same-time/same-place meeting is a defining characteristic of this type of use of a PPGIS. Clearly, all three scenarios could benefit from this type of assistance for interaction.

The second flavor of work focuses on software for *same-place/different-time*, and *different-place/different-time* meetings, e.g., as reported in Jankowski and Stasik (1996). Clearly, communication management needs to evolve to address this type of interaction. System capabilities would be needed to substitute for available human expertise. Tools that allow non-sophisticated users to perform analyses *equal* to those performed by agencies can play an important role in leveling the playing field amongst alternatives.

As discussed in NCGIA Initiative 19 (Harris and Weiner 1996), an additional important capability is to portray local or differential cultural knowledge and concepts. This knowledge, and alternatives generated, regardless of venue, must eventually interface with other schemes in the PPGIS.

The needs for and developments of PPGIS described above indicate that considerable opportunity exists for "socializing GIS". The technical developments, although important, are likely to take a back seat to the social developments of information use. Of equal importance to research on the technical capabilities of PPGIS will be the research on system use. If we do not know how information is being used in various social contexts, then our system developments might be mislead. A balanced approach to conceptual, empirical and system oriented research will likely encourage beneficial outcomes.

5. REFERENCES

Chattooga River Watershed Coalition, Southern Appalachian Forest Coalition and The Conservation Fund. 1996. *Chattooga Watershed Conservation Plan.*

Campbell, H. and I. Masser 1995. *GIS and Organizations*, Taylor & Francis, London.

Couclelis, H. and M. Monmonier. 1995. Using SUSS to Resolve NIMBY: How Spatial Understanding Support Systems can help with the "Not in My Back Yard" Syndrome. *Geographical Systems* 2:83-101.

Densham, P. J., Armstrong, M. P., and K. Kemp 1995. Report from the Specialist Meeting on Collaborative Spatial Decision Making, Initiative 17, NCGIA, U C Santa Barbara, September 17-21, 1995.

Diamond, H. and P. Noonan. 1996. *Land Use in America.: the Report of the Sustainable Use of Land Project.* Washington, DC: Island Press.

Duffy, D. M. Roseland, M., Gunton, T. I. 1996. A Preliminary Assessment of Shared Decision-making in Land Use and Natural Resource Planning, *Environments*, 23(2): 1-16.

Duwamish Coalition 1996. http://www.pan.ci.seattle.wa.us/business/dc/default.htm .

Forman, R.T. and M. Godron. 1986. *Landscape Ecology.* NY: John Wiley.

Harris, T. and D. Weiner (eds). 1996. "GIS and Society: The Social Implications of How People, Space and Environment Are Represented in GIS." Report for the I19 Meeting, March 2-5, 1996, Koinonia Retreat Center, South Haven, MN. Technical Report 96-7. UC Santa Barbara, CA: NCGIA: D7-D8.

Institute for Responsible Management 1996. Brownfields EPA Pilots News. V. 1, no.3, Dec. 15, 1996. New Brunswick, NJ.

Jankowski, P., Nyerges, T. L., Smith, A., Moore, T. J., and Horvath, E., 1997. Spatial Group Choice: A SDSS tool for collaborative spatial decision making, under review with *Intl J of Geographical Information Systems,* in press

Jankowski, P. and M. Stasik 1996. Architecture For Space And Time Distributed Collaborative Spatial Decision Making, Proceedings, GIS/LIS. Denver, CO.

Nyerges, T. 1995. Design Considerations for a Group-based GIS: Transportation Improvement Project Decision Making as an Example. *Proceedings of GIS-T '95.* Reno, NV. 261-282.

Shiffer, M. 1992. Towards a Collaborative Planning System, *Environment and Planning B*, 19:6, 709-722.

Smith, L. G. 1982. Alternative Mechanisms for Public Participation in Environmental Policy-making, *Environments* 14(3):21-34.

EXPLORING THE SOLUTION SPACE OF SEMI-STRUCTURED SPATIAL PROBLEMS USING GENETIC ALGORITHMS

David A. Bennett, Department of Geography, Southern Illinois University, Carbondale, IL 62901-4514

Greg A. Wade, Department of Computer Science, Southern Illinois University, Carbondale, IL 62901-4514

Marc P. Armstrong, Department of Geography and Program in Applied Mathematical and Computational Sciences, The University of Iowa, Iowa City, IA 52242

ABSTRACT

The resolution of semi-structured spatial problems often requires consensus building and compromise among stakeholders as they attempt to optimize their own set of criteria. The union of these sets form a criteria space that constrains the set of viable solutions that may be adopted by decision-makers. Knowledge about the criteria space, the solution space, and the relation between the two is normally incomplete and this lack of understanding places real limits on the ability of decision-makers to solve complex spatial problems. This research explores new approaches that are designed to establish a link between criteria space and solution space and to provide a mechanism that competing stakeholders can use to identify areas of conflict and compromise.

1.0 INTRODUCTION

Spatial problem solving often requires collaboration among multiple decision-makers because the effects of spatial decisions often cut across traditional bounds of discipline, jurisdiction, and ownership. Because different decision-makers will have different views of a problem, the evaluation of alternative solutions to it is complicated since: 1) a collection of spatial models and analytical tools is needed to evaluate how well each alternative meets stated criteria; 2) multicriteria evaluation tools are needed to integrate the results of these models and tools; 3) the set of all possible solutions (the solution space) is often intractable (theoretically infinite for field-based problems); and 4) not all criteria are well articulated or even known at the beginning of an analysis (*i.e.*, spatial problems are often semi-structured). Furthermore, the resolution of semi-structured spatial problems often requires consensus building and compromise among decision-makers because as individuals attempt to optimize their own set of criteria they will often come into conflict with others. The

union of these criteria sets forms a criteria space that constrains the set of viable solutions that may be adopted by decision-makers. Understanding the relation between criteria space and solution space is a key element in the successful resolution of spatial problems.

The integration of spatial decision making, spatial models and geographic information systems (GIS) has been an active area of research and advances have been made in loosely coupled systems (He *et al.*, 1995), tightly coupled systems (Bian *et al.*, 1996), and fully integrated systems (Bennett, in press; Wesseling *et al.*, 1996). At the same time researchers have been investigating techniques designed to integrate multicriteria analysis into GIS (Carver, 1991; Jankowski, 1995). What has not yet been investigated are tools to explore, analyze, and visualize the solution space of a problem with respect to multiple models and criteria. Providing such tools has several benefits: 1) new and unique solutions can be identified; 2) unarticulated criteria can be identified and incorporated into an analysis; 3) the spatial implications of specific criteria can be visualized; and 4) areas of agreement and conflict can be identified and discussed. As suggested above, the set of all possible solutions can be very large. The time required to create, model, and evaluate such large sets of possible solutions is prohibitive and heuristic tools are needed to guide and expedite this effort.

In this paper a two-dimensional genetic algorithm (Bennett *et al.*, 1996) is used to evolve landscapes that meet stated criteria based on a set of spatial models. An initial population of random landscapes is created. Each landscape is represented as a raster file in which cells are assigned a particular land cover. The fitness of a landscape is evaluated by intelligent agents that act as surrogates for decision-makers that represent competing stakeholders. Agents implement a multicriteria evaluation scheme that models the success of each landscape in meeting the stakeholders' stated criteria. Agents rank the competing landscapes and a mediating agent uses these rankings to calculate an overall fitness value for each landscape. Those landscapes deemed most "fit" by this process are used to propagate new landscapes. Thus, the solution space is heuristically expanded and explored. Delta maps derived from those landscapes that were ranked high (*e.g.*, the top three alternatives) by individual agents illustrate areas of consensus, conflict, and potential compromise. When highly ranked alternatives fail to meet the expectations of a stakeholder then the criteria space should be reevaluated.

2.0 GENETIC ALGORITHMS

Genetic algorithms are modeled after those processes that drive biological evolution and the evaluation of fitness values provides an effective heuristic for the exploration of problems that may otherwise be intractable. Alternatives in the solution space of such problems represent individuals in an evolving population. Characteristics that can be used to evaluate the relative success of individual solutions are stored in classifiers which are often implemented as bit-strings that document when a specific solution possesses a given characteristic

(Booker *et al.*, 1989; Armstrong and Bennett, 1990). Fitness in this context is proportional to how well a particular solution meets stated criteria. Three genetic operators are used to evolve a large number of new alternatives from existing alternatives: cross-over, mutation, and inversion. Mutation is a unary operator that makes random changes in a linear sequence of characteristics. Inversion, also a unary operator, flips values in a linear sequence of characteristics. Cross-over, the most powerful of these operators (De Jong, 1990), is a binary operator that generates two new offspring by duplicating two individuals (parents) and swapping "genetic code" beyond some randomly selected cross-over point. New and innovative solutions are created through random cross-overs, mutations, and inversions.

A more formal description of the genetic algorithm is as follows (after De Jong, 1990; Koza, 1994):

1. Generate an initial population, P_0, of potential solutions. These individual solutions are often created as random combinations of identified characteristics.
2. For each individual, I_m, in the current population, P_i, calculate a fitness, $f(I_m)$. Select n individuals from P_i that will be used to generate n new solutions for P_{i+1} via cross-over. The probability, p, that individual I_m will be used to create new alternatives for population, P_{i+1}, is a function of its fitness, $f(I_m)$:

$$p(I_m) = \frac{f(I_m)}{\sum_{k=1}^{n} f(I_k)} \qquad (1)$$

   where:
   $f(I_m)$ = fitness of individual m
   $p(I_m)$ = probability of individual m producing offspring in the next generation

3. Remove x individuals from P_{i+1} (based on user defined criteria).
4. Add n new individuals to P_{i+1} by applying cross-over, mutation, and inversion operators.
5. If an acceptable solution exists then stop; else advance to generation $i+1$ and return to step 2.

Although geographical applications of this approach are rare, Dibble and Densham (1993) illustrate the utility of genetic algorithms in the solution of location-allocation problems. Zhou and Civco (1996) use neural networks that employ genetic algorithms as a learning mechanism to conduct land use suitability analyses. These projects do not, however, apply genetic algorithms to two dimensional landscapes.

3.0 AGENT-DIRECTED GENETIC ALGORITHMS FOR ENVIRONMENTAL PROBLEM SOLVING

In order to manage environmental resources in privately owned landscapes it is necessary to understand how individual decisions effect environmental processes across space and through time. The tools used by resource managers to promote environmental objectives in a privately owned landscape depend largely on education, incentive-based policy initiatives (*e.g.*, conservation reserve program) and quasi-regulatory compliance programs (*e.g.*, commodity programs). Furthermore, private and public concern about the environmental ramification of land management decisions is only one of many competing issues that must be addressed by land managers. To develop feasible and politically acceptable solutions to environmental problems generated by the cumulative impact of multiple decision-makers it is often necessary to foster compromise and consensus among a diverse set of special interest groups who possess overlapping objectives; some quantifiable, and some not. Thus, environmental management, like many spatial problems, is often a semi-structured problem that requires a collaborative effort among multiple stakeholders.

3.1 Genetic Algorithms for Two Dimensional Space

Traditional genetic algorithms operate on a finite set of well-defined characteristics that are easily mapped to a linear data structure. When this approach is adapted to the generation of alternative landscapes it is necessary to extend the notion of a linear sequence of genetic code to a two dimensional representation. The linearization of space is a well-studied problem. Mark and Lauzon (1984) illustrate how to accomplish this task using a two dimensional run-length encoding (2DRE) scheme based on a Morton index of a raster-based geographical data set. Using 2DRE and two randomly selected cross-over points, two new landscapes that possess characteristics of two parent landscapes can be created (Figure 1).

3.2 Multicriteria Decision Space

The set of relevant criteria and the relative importance of specific criteria vary with the goals and objectives of the stakeholder. To integrate the concerns and objectives of multiple competing stakeholders we recast the genetic algorithm fitness function into a modified multicriteria evaluation function. To construct a composite fitness value for a given alternative and set of criteria, criterion-specific fitness values must be standardized since each analyses will not use the same units or, perhaps, even the same scale of measurement (*e.g.*, nominal, ordinal, interval, ratio). Furthermore, decision-makers must provide a subjective weighting scheme that documents the relative importance of each criterion. Several compositing schemes exist (for a review of MCE techniques in GIS see Carver 1991 and Jankowski 1995). For most of these schemes the final fitness score is a linear function that takes as input the standardized score and associated weight of each criterion.

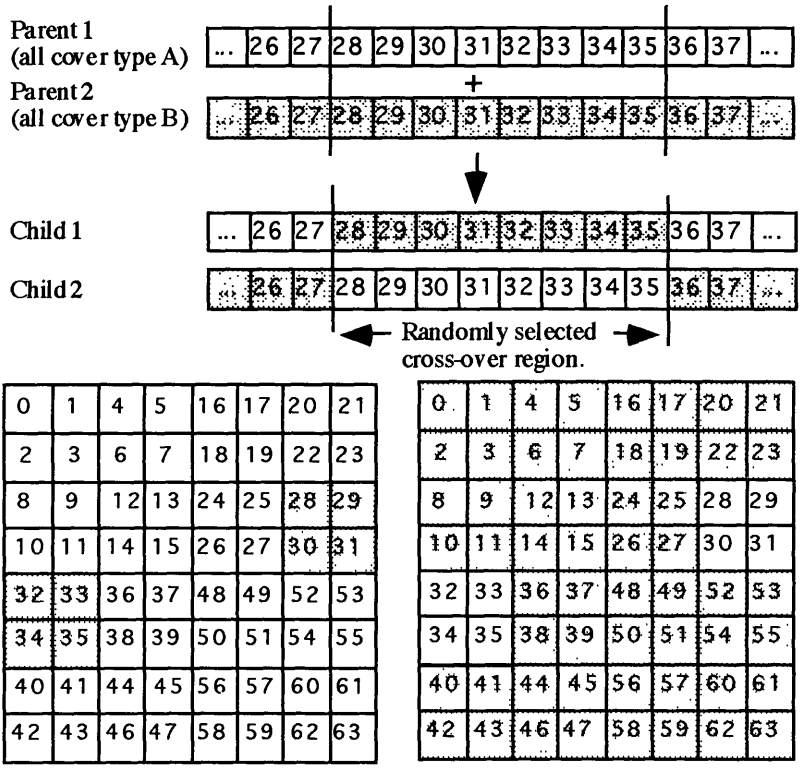


Figure 1. The results of a two dimensional cross-over.

As suggested by Breeden (1995) and determined empirically in the context of this work, the linear combination of fitness scores can be problematic in the context of genetic algorithms. If the distribution of the standardized scores for a particular criterion is positively skewed then landscapes that perform well for that criterion will be unduly favored as the next generation of solutions is created. Furthermore, outliers in the distribution of the standardized scores restrict the variance of more "typical" scores and, thus, mask potentially significant differences among alternative solutions. As a result, the probability that an individual landscape will be propagated into the next generation will be approximately equal for many individuals in the population. One way to overcome the impact of outliers and skewed distributions is to construct a composite fitness value based on ranked order. The fitness value for a particular landscape is then calculated as (Breeden, 1995):

$$f(I_{kd}) = \sum_{j=1}^{n} w_{jd} r_{kjd} \qquad (2)$$

where:

r_{kjd} = rank score of alternative k given criterion j and decision maker d.

w_{jd} = weighted value of criterion j given decision maker d.

$f(I_{kd})$ = fitness value for alternative k given decision maker d.

Criteria weights reflect a qualitative assessment of a single decision-maker, a class of decision-makers or, perhaps, a set of decision-makers who have reached a consensus. If a consensus has been reached then, in many situations, compromise among competing decision-makers has been reached in criteria space. This may or may not be possible. An alternative approach is to allow individual decision-makers, or sets of decision-makers that represent specific classes of stakeholders, to define criteria independently and attempt to construct a compromise in the solution space. To accomplish this using a genetic algorithm a global fitness value for each landscape must be calculated to represent all decision-makers. Here this global fitness value is calculated as the mean of the independent fitness values:

$$F(I_k) = \frac{\sum_{d=1}^{n} f(I_{kd})}{n} \qquad (3)$$

where:

$F(I_k)$ = Global fitness value for individual k.

$f(I_{kd})$= Fitness value for individual k given decision-maker d.

n = Total number of decision-makers.

3.3 Intelligent Agents

As geoprocessing software becomes more sophisticated it is able to support the analysis of an increasingly rich set of problems. This richness, however, has a downside: software has become increasingly complex and, thus, more difficult to use. Furthermore, decision-makers often represent several interests and bring to the negotiation table different types of training, levels of education, experience with computing technologies, and familiarity with the problem that is being addressed. Though such differences can prove valuable since distributed expertise may allow for decision making procedures that are less prone to errors attributable to a lack of domain specific knowledge, this differential in knowledge can also have interaction effects that complicate the decision making process. Because of the number of analytical tools available and the disparate backgrounds of individual decision-makers, users may not always understand the implications of particular analytical methods. In many cases, additional knowledge may be required to support informed use.

One way to provide a more common level of support to decision-makers is to create intelligent software agents equipped with knowledge about how and when to implement specific analytical tools. At this point, two classes of intelligent agents have been implemented, mediating agents and user agents (see Shoham, 1993 for a discussion on intelligent agents). User agents acting on behalf of specific decision-makers, calculate $f(I_{ka})$ (equation 1) for each individual landscape k using applicable analytical tools and user supplied

criteria weights, and returns these fitness values to the mediating agent. Using this information the mediating agent calculates $F(I_k)$ (equation 2), selects individuals for propagation and builds consensus among competing interests.

4.0 A CASE STUDY

A multidisciplinary research team from Southern Illinois University at Carbondale is investigating the impact of alternative resource policy and management scenarios on the economy, hydrology, and ecology of the Cache River (IL) watershed. The goal of this research effort is to develop a land use management plan that is generally acceptable to all stakeholders. A small study site within this watershed was selected to test the utility of two-dimensional genetic algorithms in the resolution of semi-structured spatial problems. The study site is approximately 3.69km^2 captured as a grid with a 30m cell resolution (64 rows and 64 columns). This site was selected because it possesses considerable spatial variability within a manageable area. Alternative landscapes are comprised of corn, soybean, double crop (winter wheat then soybean), wheat, grassland, and forest.

Stakeholders within the region were generalized into three classes:

1. Farmers who want to maximize farm revenue.
2. Conservationists interested in reducing soil loss and non-point pollution and agricultural productivity.
3. Wildlife enthusiasts, local entrepreneurs, and recreational hunters interested in the maintenance and enhancement of wildlife populations.

To assess how well alternative landscapes meet the concerns of these stakeholders models were developed that evaluate agricultural income, soil erosion, and the interspersion and juxtaposition of land cover types. To support these models spatial databases were developed that capture the topographic and edaphic characteristics of the study area.

Agricultural income is generated from corn, soybean, wheat, and hay (grasslands). No income is attributed to forest land. For each alternative landscape net agricultural return is calculated for each 30x30m cell by considering land cover, the expected productivity of that cover type given the associated soil, the market value of that crop, and the expected costs of producing that crop. Market prices and production costs for agricultural produce are based on ten year averages for the state of Illinois. Soil productivity values are derived from the Union County, IL soil survey (USDA, 1979). An estimate of the rate of erosion that is associated with each cell is calculated using the universal soil loss equation. The estimated value for soil erodibility (K) was derived from the Union County soil survey. The cropping factor (C) is an estimate based on cover type. A 7.5 minute DEM was used to estimate the slope of each cell. This information was, in turn, used to estimate the LS factor of the universal soil loss equation. Land management practices (P) were assumed to

be the same on all cells. Interspersion is an index of the "intermixing of units of different habitat types" (Giles 1978:156). It is assumed that interspersion is desirable for wildlife but can lead to inefficiencies in agricultural production. Juxtaposition, as used here, is a measure of adjacency among cover types. The value of an edge between land cover types depends on the objectives of the land manager and the cover types involved.

A user agent was created to represent each stakeholder class. Each agent maintains a weight and a ranked list of alternatives for each criteria/model. Criteria weights used by each agent are listed in Table 1. An attempt is made to maximize all criteria except soil loss which is minimized. Note that these values were used only for "proof-of-concept" and, thus, are not intended to be representative of the groups identified.

	Ag. Production	Soil Loss	Interspersion	Juxtaposition
Farmer	1	0	0	0
Conservationist	0.5	0.5	0	0
Wildlife Enthus.	0.25	0	0.5	0.25

Table 1. Agent Weights

Figure 2 illustrates the two most influential landscape characteristics, soil erodibility and soil productivity. The dominant landscape feature within the study area is a floodplain that runs from the northeast to the southwest. A somewhat smaller tributary enters the study area in the northeast corner and continues south until it meets the larger floodplain. As can be seen in Figure 2A, the side slopes of these valleys are highly erodible and the uplands are moderately erodible. Most of the highly productive soils are located within the floodplain of the two streams (Figure 2B). However, in the southwest the floodplain is too wet to provide a reliable crop. An initial set of random landscapes were created and this "population" was allowed to evolve for more than 200 generations. The results of this experiment are presented in Figure 3. The spatial patterns that evolved through this process are logical given the character of landscape and represent reasonable compromise solutions given the objectives of the stakeholders (highly erodible and low producing soils in forest, moderately erodible soils with reasonable productivity in wheat, slightly erodible productive soils in soybeans).

5.0 CONCLUSION

To support effective resource management practices new tools are needed that allow decision-makers to build consensus among multiple stakeholders and to investigate the cumulative impact of individual actions. This research investigates two technologies that offer promise for such collaborative spatial decision making processes, agent-oriented programming and genetic algorithms. Genetic algorithms are used here to evolve landscapes that meet predetermined criteria. Intelligent agents provide a means of evaluating the fitness of these landscapes based on weighted criteria. Through this interaction between

intelligent agents and genetic algorithms management strategies can evolve in ways that begin to meet the goals of multiple stakeholders.

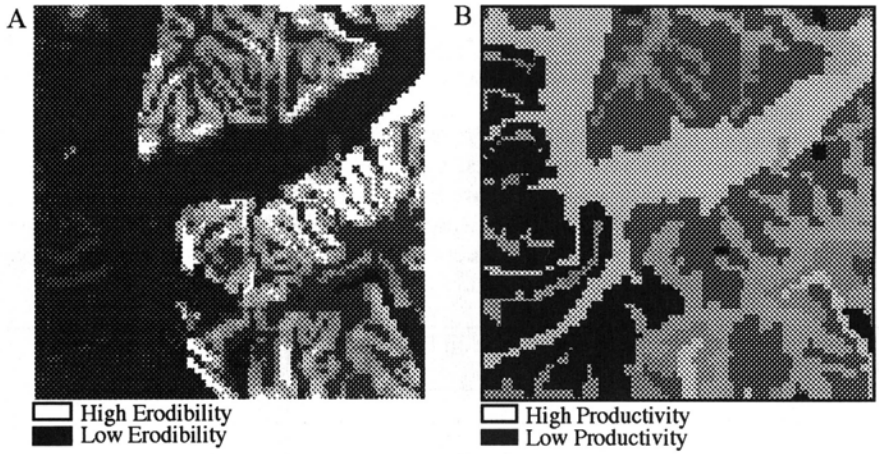


Figure 2. Soil erodibility and agricultural productivity maps for the study site.

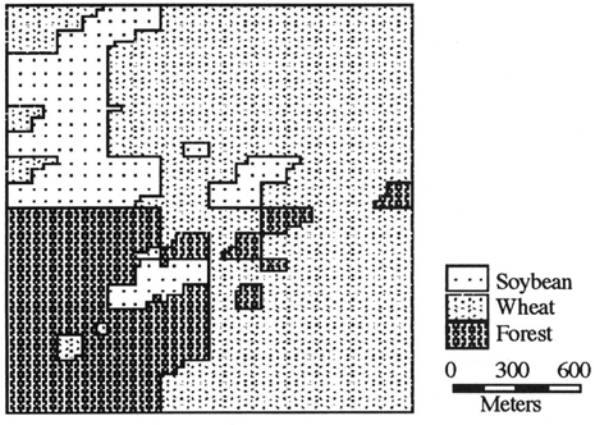


Figure 3. Most "fit" landscape after 200 generations.

6.0 REFERENCES

Armstrong, M.P. and Bennett, D.A. (1990). A bit-mapped classifier for groundwater quality assessment. *Computers and Geosciences*, 16 (6): 811-832.

Bennett, D.A. (in press). A framework for the integration of geographic information systems and modelbase management. *International Journal of Geographical Information Systems*.

Bennett, D.A., Armstrong M.P. and Wade, G. A. (1996). Agent mediated consensus-building for environmental problems: A genetic algorithm approach. *Third International Conference on Environmental Modeling and Geographic Information Systems*, Santa Barbara, CA: National Center for Geographical Analysis, CD and WWW.

Bian, L., Sun, H., Blodgett C., Egbert, S., Li, W., Ran, L., and Koussis A. (1996). An integrated interface system to couple the SWAT model and ARC/INFO. *Third International Conference on Environmental Modeling and Geographic Information Systems*, Santa Barbara, CA: National Center for Geographical Analysis, CD and WWW.

Booker, L.B., Goldberg, D.E., and Holland, J.H. (1989). Classifier Systems and Genetic Algorithms. *Artificial Intelligence*, 40, pp. 235-282.

Breeden, J.L. (1995). Optimizing stochastic and multiple fitness functions. In *Evolutionary Programming*, edited by McDonnell, J.R., Reynolds, R.G., and Fogel, D.B. MIT Press, Cambridge, MA.

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

De Jong, K. (1990). Genetic-algorithm-based learning. In *Machine Learning* edited by Y. Kodratoff and R. Michalski. Morgan Kaufmann, San Mateo, CA.

Dibble, K., and Densham P.A. (1993). Generating interesting alternatives in GIS and SDSS using genetic algorithms. In *Proceedings of GIS/LIS '93 , Volume 1*. Bethesda, MD: American Congress on Surveying and Mapping, pp. 180-189.

Giles, R.H. (1978). *Wildlife Management*. Freeman, San Francisco, CA.

He, C., J.F. Riggs and Kang, Y.T. (1993). Integration of geographic information systems and a computer model to evaluate impacts of agricultural runoff on water quality. *Water Resources Bulletin* 29, pp. 891-900.

Koza, J.R. (1994). Introduction to genetic programming. In *Advances in Genetic Programming* edited by Kinnear, K.E. MIT Press, Cambridge, MA.

Jankowski, P. (1995). Integrating geographical information systems and multiple criteria decision-making methods. *International Journal of Geographical Information Systems*, 9(3), pp. 251-275.

Mark, D.M., and Lauzon, J.P. (1984). Linear quadtrees for geographic information systems, *Proceedings of the International Symposium on Spatial Data Handling*, Zurich, pp. 412-430.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60, pp. 51-92.

USDA (1979). *Soil Survey of Unions County, Illinois*. National Cooperative Soil Survey.

Wesseling, C.G, Karssenberg, D., Burrough, P.A., and van Deursen, W.P. (1996) Integrating dynamic environmental models in GIS: The development of a dynamic modelling language. *Transactions in GIS*, 1(1), pp. 40-48.

Zhou, J. and Civco, D. (1996). Using genetic learning neural networks for spatial decision making in GIS. *Photogrammetric Engineering and Remote Sensing*, 62(11), pp. 1287-1295.

A PUBLIC PARTICIPATION APPROACH TO CHARTING INFORMATION SPACES

Paul Schroeder
Doctoral Student
Dept. of Spatial Information Science and Engineering
University of Maine, Orono, Maine USA

ABSTRACT

This paper explores the use of GIS concepts to model information access barriers. This potential application is considered in light of recent meetings centered on the social impacts of GIS and its use in public participation settings and new community institutions. Models of the information access concept are presented, with other attempts to chart or map elements of information infrastructures, environments and spaces. The notion of a mixed delivery network / social network model is presented. The co-production of such a model in the setting of a state network is outlined. The paper concludes with a reflection on the meaning of "home" and the subjective sense of spaciousness in information environments.

INTRODUCTION

Several recent meetings sponsored by the National Center for Geographic Information and Analysis (NCGIA) were an opportunity for the GIS research community to join in a broadening conversation about the future of the digital technologies that have been under intensive construction for the past 50 years.

Broadly stated, the following questions were raised: Can the maturing spatial technologies, beginning with GIS, be brought to bear fruitfully on the problems of the geographies of knowledge that have surfaced through the appearance of globally networked personal computers? What freight do present technical approaches bring into this domain? How much real and useful information is being lost through the dominant focus on digital technologies? And, as the technical apparatus as we know it is only marginally satisfactory from some perspectives (and a downright threat from others), what can be done to right the situation in the future?

As a librarian concerned with public information resources, I brought to the meetings a notion that information access barriers could be represented through some form of GIS. Developing this notion as informed by the GIS and Society meeting and Public Participation GIS Workshop is the theme of this paper.

NCGIA INITIATIVE 19 and PUBLIC PARTICIPATION GIS

A conceptual thread that was heard frequently at the Initiative 19 *GIS and Society* specialists' meeting (March 1996) was discussion of the potential for a "GIS 2." Opening remarks by Michael Curry on spatial data institutions framed his question: What might GIS be? Critical perspectives on present GIS were shared by many at the meeting. The data-driven nature of present technology, its origins in the requirements of government administration, its adoption for surveillance and mass marketing purposes, its gender bias and roots in Cartesian thought were all vigorously discussed.

These views were balanced by a generally optimistic vision that some forms of GIS, versions of "GIS 2," could be developed and put to work toward goals of social and environmental justice. The discussion of GIS 2 accompanied discussion of "geographies of virtual worlds," pointing at the convergence of GIS with the potentials of widespread data communications networks.

A summary document of this discussion, "Criteria for the Design of a GIS 2," was presented at the concluding session of the I-19 meeting. These criteria envisaged systems would allow expanded end-user contributions of information, would integrate diverse media including sketch maps and audio, and would provide better handling of change over time. In sum, future systems should strive to preserve local knowledge while relaxing formal constraints such as those of the system of Cartesian coordinates.

Discussion of the GIS 2 theme was continued at the Public Participation GIS Workshop (July, 1996), with focus on improved integration of GIS technology into the domain of public involvement solution of public policy controversies. The Workshop was structured to promote discussion of topics related to technical as well as public process issues.

Environmental and natural resource issues and applications in urban and regional planning were the settings in which participatory GIS was largely discussed. Presentations were made on the following aspects: dispute resolution, urban data resources, neighborhood organizing, spatial conflict, the integration of multimedia and GIS, object-oriented computing, and future technologies such as "smart" compasses, binoculars and maps.

The PPGIS Workshop discussions tended toward two poles of emphasis: the development of new spatial technologies without direct reference to a specific applications domain, versus concern for what will be required of future technology in support of the human interactions that are central to enabling settling public policy disputes. The five original "Criteria" for a GIS 2 were expanded to thirteen. Needs for specific capabilities were noted, such as tools to enable discovery of precedents globally for local problem situations.

Publications that stand as examples of the themes of this Workshop include Aberle (1993), Bunge and Bordessa (1975), Couclelis and Monmonier (1995), Dunn and Lichtenstein (1994), Schmitt and Brassel (1996), and Shiffer (1995). (See Note *)

CREATING NEW COMMUNITY INSTITUTIONS

Two distinct points of emphasis, on technologies and on public processes, intersect in discussion of the physical location of a GIS 2. Would this be found in specific places, or would these tools largely be put to work in distributed network space? This points toward the mutual importance of people, technologies and geography in consideration of the design of information systems and in representing them.

The notion of a "social learning center" or "community learning center" arose in response to assumptions that a place will always be needed, a real physical place within communities, where the activities supported by public participation technologies will go on. This new community institution would be built on the base already in place in the form of public institutions such as schools and libraries. Such a place would also require staffs trained in new professional roles, combining in some measure the teacher, librarian, technology coordinator and mediator. Without devoting resources toward developing such centers, the potential of a "public participation GIS" would not likely be realized.

Such centers have frequently been included within advanced technology scenarios. A contemporary futuristic vision is "tele-immersion" as propounded by the Internet 2 Applications Group, in which "high speed communications systems support collaboration applications in multiple geographically distributed immersive environments" (Internet 2, 1997). An earlier proposal, less technology-dependent, can be recalled in Brun's "houses of heuristics" (Brun, 1974).

---

* For background to NCGIA I-19 *GIS and Society* see: Pickles (1995), Harris and Weiner (1996) and Sheppard and Poiker (1995). Papers and reports from meetings mentioned here are available at the following sites: Public Participation GIS Workshop, at http://ncgia.spatial.maine.edu/ppgis/ppgishom.html; GIS and Society, NCGIA Initiative 19, at http://www.geo.wvu.edu/www/i19/page.html; Collaborative Spatial Decision Making, NCGIA Initiative 17, at http://www.ncgia.ucsb.edu/research/i17/I-17_home.html; Formal Models of Common Sense Geographic Worlds, NCGIA Initiative 21, at http://ncgia.geog.buffalo.edu/ncgia/i21.html; Spatial Technologies, Geographic Information, and the City, at http://www.ncgia.ucsb.edu/conf/BALTIMORE/menu.html; Theories and Metaphors of Cyberspace, at http://pespmc1.vub.ac.be/cybspasy.html

Considering prototypes for such a facility leads directly to the complex relations of people, machines, places and information environments in which they will be placed.

INFORMATION SEEKING AS SOCIETY-BUILDING

In some respects the themes of PPGIS parallel the themes of NCGIA Initiative 17: Collaborative Spatial Decision Making (see Note *), with the following distinction: I-17 conceives of a form of collaborative group work, while PPGIS considers the more general case of open public involvement. The problem of charting information spaces seems well suited to take advantage of this inclusive public involvement approach.

A framework may be set by considering information seeking as the ad hoc creation of a society. The simplest case might be that of asking directions in public space. A new and temporary society of two is created. That there is a geographic component in this situation is clear.

Extension of this simple case points toward the widespread use of wayfinding and navigation metaphors for information seeking in general (for example see Canter et al. 1985). A more complex case is seen in the creation of a paper for a conference such as this. The ad hoc society is represented in the paper's references, which stand both as a subset of information sources and as a set of pointers for readers and listeners. The craft of bibliography, within the area of the librarian's expertise, shares something with the craft of cartography in that references are the landmarks of the bibliographic search. In today's expanding but unstable information environment, the arts of reference (see McArthur 1986) are an area in need of development. For spatial data these are the metadata questions.

Wayfinding's simple "society of two" points toward a potentially paradoxical condition: the questioner enters into, and actually creates, the domain of her or his own observations. This implies a different process than is assumed in the fact-finding of traditional science, in which the observer is excluded from the domain of he observations. Accounting for this necessary condition is a task set out by "second-order cybernetics" (Brier, 1992).

Congenial public spaces could be rated according to the relative ease with which a stranger may be approached with a wayfinding question. By extension, information systems could be evaluated in terms of their affording of the abilities to speak, beginning with the range of questions legitimately allowed (see Chatman, 1996). Capable systems allow the articulation of diverse voices.

CHARTING ACCESS BARRIERS TO INFORMATION RESOURCES

As with navigation metaphors, the concept of "access" is often introduced into discussions of public information systems. Access implies open and closed spaces in which information somehow resides, and a sort of directionality from here to there. The access concept is tied to the notion that information is an object that can be retrieved.

The access notion also supports economic views of information that emphasize costs and consider information as a form of commodity. I began from this customary framework in asking whether we might learn something about patterns and problems in information access from the application of spatial models similar to those found in GIS. Could something like a friction surface map be generated that might represent the places where information lies, the points of origin from which questions are asked, and the relative cost gradients of various pathways between these? Such a model could potentially be used by librarians, for example, in making choices among available information resources, and by public policy planners in devising suitable public access arrangements.

An outline of how similar tools could be put to use in the context of physical accessibility problems in urban environments has been put forward by M. Gould for the *Spatial Technologies, Geographic Information and the City* conference (see Note *).

This approach applied to information access issues will break down because of general limitations on the information-as-object model. Braman (1995) provides detailed discussion of the tangible versus the intangible qualities of information in the context of applying economic models to information resources. (For further discussion of information as object see Reddy, 1993).

Relating three concepts may be helpful at this point: information infrastructures, environments, and spaces. Consider infrastructure to be the places where the tangible aspects of information are established. The information environment is where the intangible aspects, equally important but less measurable, are to be found. And information space joins the two, conceived either as an envelope or as an intersection, perhaps Schatz' "interspace" (1995).

In these terms, information infrastructure would include telecommunications networks, the contents of public libraries, the policies and costs related to their operations, and similar phenomena. Information environment would pertain to the social structures surrounding the infrastructure (as seen in the ad hoc societies noted above), and would need to account for the emergence of questions themselves and the criteria by which answers are accepted and judged. The concept of information space would be inclusive of both of these aspects.

How would access barriers be treated in a comprehensive model that seeks to account for the entire information space, including the intangible elements? If information seeking is conceived as initiating a conversation within the information environment, then access would be linked to the relative inclusiveness or exclusiveness of the conversational choices available to participants. Tools such as Paulston and Liebman's "social cartography" (1993) would find a place in such a representation and analysis of the information environment. Kochen (1989) includes many studies involving social network maps.

The initial topographic image of access barriers, featuring costly paths between open and closed information spaces, could be supplemented by a topological approach representing pure boundaries without bounded areas or spaces.

RELATED APPROACHES TO INFORMATION SPACE MODELING

This section presents several approaches to representing the tangible infrastructure and intangible environment of information. The most comprehensive attempt at the former is the set of "information delivery network" maps of the Pacific Northwest presented by Murr et al. (1985). A similar recent example was produced by Williams (1995), mapping distributions of libraries with Internet access and other public information institutions in South Carolina. An example portraying diffusion of libraries and newspapers in 19th century Canada is given by Wadland and Hobbs (1993). Also in this tradition are the charts of Internet diffusion presented in the Matrix Mapping Quarterly (Matrix in the World, 1995).

The project of charting the social environment of information is presented in great detail in the *Encyclopedia of World Problems and Human Potentials* (Union of International Associations, 1991). Thousands of "human problems" are cataloged and cross-referenced, including over 100 examples amounting to a typology of obstacles in information space. Detailed requirements for "mapping social complexity" are provided. Another typology of information barriers at much smaller scale (five classes) is presented by Brown (1991). From the community of GIS researchers, the approach of Lemberg (1996) shares many elements with suggestions presented here.

The notion of charting access also has been explicitly suggested by Mitchell (1995, p. 131) with reference to G. Nolli's 18th century maps of Rome showing public and private spaces: "Perhaps some electronic cartographer of the future will produce an appropriately nuanced Nolli map of the Net."

The existence of these related efforts at least points to a perceived need for modeling or creating conceptual frameworks in this area. A possible application area is described in the next section.

APPLICATIONS IN A STATE NETWORK SETTING

Assuming that a satisfactory model could be devised along the lines described here, who would construct, maintain and use charts of public information spaces? I foresee that information policy planners, educators, public librarians, transportation planners and others with similar professional responsibilities could co-produce and use such tools.

The state level is a unit of geographic importance for the development of public information resources. Even under deregulation state policy will continue to guide telecommunications developments as under previous regulatory frameworks. In Maine, a 1,000-site school and library data network is under construction. Maine has looked to the "North Carolina" network as a model, as well as to the "Kansas" and "Georgia" models for state information access policies.

The institutions connecting to the networks certainly view those connections spatially. This new information space represents hazards similar to those assumed in unfamiliar physical terrain. School administrators are uncertain about their abilities to provide safe information space in this new information environment.

The need to convey a useful image of the network and its potentials has become a priority of the Maine Internet Education Consortium, which has assumed responsibility for end-user training throughout the state. It was recently reported that only two of the state's 150 school superintendents regularly use e-mail. This nonparticipation is attributed in part to lack of coherent images of the network and its possibilities.

The Consortium's Board has assembled a workgroup to develop "maps and tools" aimed at informing school administrators about the network. It is foreseen that these would ultimately be created collectively by the connected sites, and would be used for identifying projects, sharing expertise, and locating information resources regionally. This could be done online, and could begin with the site and district base maps already completed by the state's Office of Geographic Information Services. With a pervasive network in place, the actual construction of a widely distributed system model will be possible.

AT HOME IN A SPACIOUS INFORMATION ENVIRONMENT

The project of cooperative construction of charts of information spaces built using the capabilities of future spatial technologies (or Mitchell's "Nolli maps of the Net") may never be fully realized. Nonetheless, models of those spaces will by analogy serve to cast light on the complexities of the public information environment.

Today, many have "home pages" on the Web. How often do these convey the feeling of home? The domed reading rooms of the British Library and Library of Congress, and other library settings in smaller scale, produce a sense of spaciousness surrounding the information quest. Though the world of the Web claims to afford wider worlds of access, online searching still seems subjectively to proceed within a severely confined space.

Among the challenges facing such efforts as GIS 2 and its counterpart Internet 2 will be the building of a spacious information environment in which many more of us can find a home and voice.

ACKNOWLEDGMENTS

I would like to express appreciation to the NCGIA for supporting my participation at the two meetings discussed here. Thanks to Harlan Onsrud, Kate Beard, Max Egenhofer and Xavier Lopez for conversations on the themes of this paper. My graduate studies have been supported by fellowships from NCGIA and the University of Maine.

REFERENCES

Aberle, D., ed. (1993). *Boundaries of Home: Mapping for Local Empowerment.* Gabriola Island, BC, New Society.

Braman, S. (1995) "Alternative Conceptualizations of the Information Economy." In, *Advances in Librarianship* 19 (1995), ed. by I. Godden, pp. 99-116. San Diego, Academic Press.

Brown, M.E. (1991). "A General Model of Information-Seeking Behavior." In, *Proceedings, ASIS '91*, pp. 9-14. Medford, NJ, Learned Information.

Brier, S. (1992). "Information and Consciousness: A Critique of the Mechanistic Concept of Information." *Cybernetics and Human Knowing* 1(2/3):71-94.

Brun, H. (1974). "The Need of Cognition for the Cognition of Needs." In, *Cybernetics of Cybernetics*, ed. by H. Von Foerster, pp. 336-341. Biological Computer Laboratory Report 73.38, Urbana, IL.

Bunge, W.W., and R. Bordessa. (1975). *The Canadian Alternative: Survival, Expeditions and Urban Change.* Toronto, Department of Geography, York Univesity.

Canter, D., R. Rivers and G. Storrs. (1985). "Characterizing User Navigation Through Complex Data Structures." *Behaviour and Information Technology* 4(2):93-102.

Chatman, E.A. (1996). "The Impoverished Life-World of Outsiders." *Journal of the American Society for Information Science* 47(3):193-206.

Couclelis, H., and M. Monmonier. (1995). "Using SUSS to Resolve NIMBY: How Spatial Understanding Support Systems Can Help With the 'Not In My Back Yard' Syndrome. *Geographical Systems* 2(2):83-101.

Dunn, W.T., and E.C. Lichtenstein. (1994) *Public Participation in Environmental Decisionmaking*. Washington, DC, Division for Public Services, American Bar Association.

Harris, T., and D. Weiner, eds. (1996). *GIS and Society: The Social Implications of How People, Space and Environment Are Represented in GIS*. Scientific Report for the Initiative 19 Specialist Meeting, (South Haven, MN, March 2-5, 1996). NCGIA Technical Report TR 96-7. Santa Barbara, CA, National Center for Geographic Information and Analysis. [Also available online, see Note *]

Internet 2 Applications Working Group. (1997). *Internet 2 Applications and Applications Framework*. Version of Jan. 1. At, http://www.unc.edu/~whgraves/i2-apps.html

Kochen, Manfred, ed. (1989). *The Small World*. Norwood, NJ, Ablex.

Lemberg, D. (1996) "Gentleman Adventurers in the Information Age: Accessibility, Activity, and Urban Futures." Position paper for *Spatial Technologies, Geographic Information and the City*. [See Note *].

The Matrix in the World, July 1995. (1995) Map, 1:65 million, Winkel Tripel Projection. Austin, TX, Matrix Information and Directory Services.

McArthur, T. (1986) *Worlds of Reference: Lexicography, Learning and Language from the Clay Tablet to the Computer*. Cambridge, Cambridge University Press.

Mitchell, W. J. (1995). *City of Bits: Space, Place and the Infobahn*. Cambridge, MA, MIT Press.

Murr, L.E., J.B. Williams and R.-E. Miller. (1985). *Information Highways: Mapping Information Delivery Networks in the Pacific Northwest*. Portland, OR, Hypermap.

Paulston, R.G., and M. Liebman. (1993). "An Invitation to Postmodern Social Cartography." Paper presented at the *Annual Meeting of the Comparative International Education Society* (37th, Kingston, Jamaica, March 1993). ERIC Document ED 358 576

Pickles, J., ed. (1995). *Ground Truth: The Social Implications of Geographic Information Systems*. New York, Guilford Press.

Reddy, M.J. (1993). "The Conduit Metaphor: A Case of Frame Conflict In Our Language About Language." In, A. Ortony, ed., *Metaphor and Thought*, 2nd ed., pp. 164-201. Cambridge, Cambridge University Press.

Schatz, B.R. (1995) "Information Analysis in the Net: The Interspace of the Twenty-First Century." Keynote Plenary Lecture, *American Society for Information Science Annual Meeting*, Chicago, October 11, 1995. http://csl.ncsa.uiuc.edu/IS.html

Schmitt, E., and K. Brassel. (1996) "From GIS for Control to GIS for Creative Exploration." In, Harris and Weiner, pp. D61-D63.

Sheppard, E., and T. Poiker, eds. (1995). "GIS and Society." Special content issue. *Cartography and Geographic Information Systems* 22(1) (January, 1995).

Shiffer, M.J. (1995) "Interactive Multimedia Planning Support: Moving from Stand-Alone Systems to the World Wide Web." *Environment and Planning B: Planning and Design* 22:649-664.

Union of International Associations. (1991) *Encyclopedia of World Problems and Human Potentials*. 3rd ed. Vol. 1: World Problems. Vol. 2: Human Potentials. Munich, K.G. Saur.

Wadland, J.H., and M. Hobbs. (1993). "The Printed Word." In, *Historical Atlas of Canada*, ed. by R. Louis Gentilcore, Vol. 2: The Land Transformed, 1800-1891, Pl. 51. Toronto, University of Toronto Press.

Williams, R.V. (1995). "Mapping and Measuring the Information Infrastructure for Planning Purposes: Preliminary Study of South Carolina." In, *Proceedings, ASIS '95*, pp. 144-151.

GIS, SOCIETY, AND DECISIONS: A NEW DIRECTION WITH SUDSS IN COMMAND?

T.J. Moore, Department of Geography
University of Washington, Box 353550, Seattle, WA 98195-3550
tjmoore@u.washington.edu

Geographic information systems (GIS) have become easier to use and very popular in recent years. However, many users are finding that while the technology currently provides an impressive and expanding toolbox for scientific and technical analyses to support problem-solving, the application of GIS in group decision-making contexts -- particularly in the public sphere -- tends to identify weaknesses in the technology. This paper proposes a vision for a new kind of GIS for an altogether different role: societal decision-making. Following the evolution of terminology in the literature, the name proposed for this class of technology is "Spatial Understanding and Decision Support Systems" (SUDSS). The name offered for that class of activity in which the technology may be applied is "COllaborative Mapping, Modeling, and Analysis for Negotiation and Decision-making" (COMMAND). The paper provides a very brief introduction to the recent developments within society, and within the research literature, which make this proposal a timely one. Then, an overarching framework is outlined, and possible features of a SUDSS are presented. A particular focus of this discussion is centered on potential design components that may facilitate group discussion distributed over space and time. The paper concludes with a short discussion concerning why the underlying design philosophy may be an important change for the discipline of geography and for society.

INTRODUCTION

Problems like the NIMBY (Not-In-My-BackYard) Syndrome have caused the long-term delay, or outright cancellation, of many proposed projects with locally undesirable impacts because, in part, of the sense that the decision-making process has been removed, historically, from the hands (or voices) of those about to receive the newly proposed burdens (e.g., see Couclelis and Monmonier, 1995). So, in the sphere of public policy in the United States and elsewhere, "public participation" and "community visioning" are becoming more widely accepted as potential ways to promote a more inclusive decision-making process. However, is GIS technology sufficiently designed for societal decision-making tasks?

Great strides have been made toward improving the power and user-friendliness of GIS technology for a wider audience. Furthermore, spatial databases are quickly growing in size. Nevertheless, as GIS technology matures, researchers, software developers, and software users (including technical specialists and decision-makers) are beginning to better understand that their technology may not be sufficiently designed to openly facilitate decision-making, particularly at the group-level. Therefore, the GIS research community spawned a number of research initiatives directed at exploring elements of the role of this technology in society and decision-making (e.g., National Center for Geographic Information and Analysis [NCGIA] Research Initiatives 6, 17, 19, 20, and 21).

This paper presents a vision of a new fold of technology, called Spatial Understanding and Decision Support Systems (SUDSS), which may help users develop a sense of shared understanding about a problem *and* the possible solutions, all while the participants are distributed over space and time. The kind of activities that SUDSS can be applied to are collectively called COllaborative Mapping, Modeling, and Analysis for Negotiation and Decision-making (COMMAND).

THE RESEARCH CONTEXT

The contributions to the literature on this subject have been growing rapidly (e.g., Armstrong, 1994; Carver, 1991; Couclelis and Monmonier, 1995; Densham, 1991; Faber *et al.*, 1994; Heywood and Carver, 1994; Jankowski, *et al.*, 1997; Malczewski 1996; Nyerges, 1995; Nyerges and Jankowski 1994, 1997; and Shiffer 1992, 1995), partly as a result of research initiatives sponsored by the NCGIA.

The SUDSS acronym proposed here continues to build on recent directions in the literature. Couclelis and Monmonier (1995) have called for an extension of spatial decision support system (SDSS) tools into the realm of "spatial understanding," in what they refer to as spatial understanding support systems (SUSS). The authors justifiably note that current GIS/SDSS tools do not provide support for problem exploration and the generation of shared understanding among diverse stakeholders. Using NIMBY public-policy problems as the canvas for their work, the authors then began their inquiry into possible alternative software functionality which might yield the "narratives" of social context that seem to be so desperately needed. Heywood and Carver (1994) refer to this missing element as an "idea generation system." Whatever the name, the call is clear: the technology requires a design which allows the users more time to creatively and flexibly explore spatial data and their

relationships, and then reflect upon, and discuss, this information at a group level.

EXPLORING SUDSS

Framework

Figure 1 presents a generalized framework of the modular components of a SUDSS. These components are categorized into three groups: 1) discussion group (issue management subsystem, or IMS), 2) geographic group (geographic information subsystem), and 3) quantitative and qualitative decision-making group (with the group aiding, process modeling, and decision modeling subsystems). The discussion group is composed of the issue management subsystem, which will be a primary focus of the remainder of this paper. The geographic group (geographic information subsystem) contains tools such as: database management, spatial analysis, data presentation, a data dictionary, and a data quality filter (see, for example, Paradis and Beard, 1994). The decision-making group is composed of decision modeling tools (e.g., multicriteria decision techniques for choice tasks), process modeling tools (simulation modeling of, e.g., environmental, economic, and social conditions; sensitivity and uncertainty analyses), and group aiding tools (such as group voting resources; a group whiteboard).

This framework is presented in Figure 1, with a general schema organized to reflect that the IMS and the GIS are the central players shaping the fundamental features of a SUDSS environment. The qualities of the other subsystems are more likely to be different across applications, depending upon the needs of the users.

Overview of an IMS

The IMS is envisioned to provide the core functionality to allow group members to participate in a somewhat structured, distributed, and asynchronous computer-supported discussion devoted to spatial understanding, alternative generation, and alternative selection. The subsystem will provide users with the ability to contribute "nodes" to a running conversation which address matters of interest and concern. This computer-supported conversation could be organized by the users into a set of "discourse maps" on separate, but broad, topics of relevance to the overall decision at hand. These discourse maps can be viewed, edited, queried, and archived. To query a discourse map, the participants could use geographic location, location within the discussion, aspatial attributes (e.g., topic keywords), decision task context, time (world, database, or discussion), discussant identification information, node type (i.e., rhetorical construct), or document type (map, text document, etc.). For each contribution to the on-

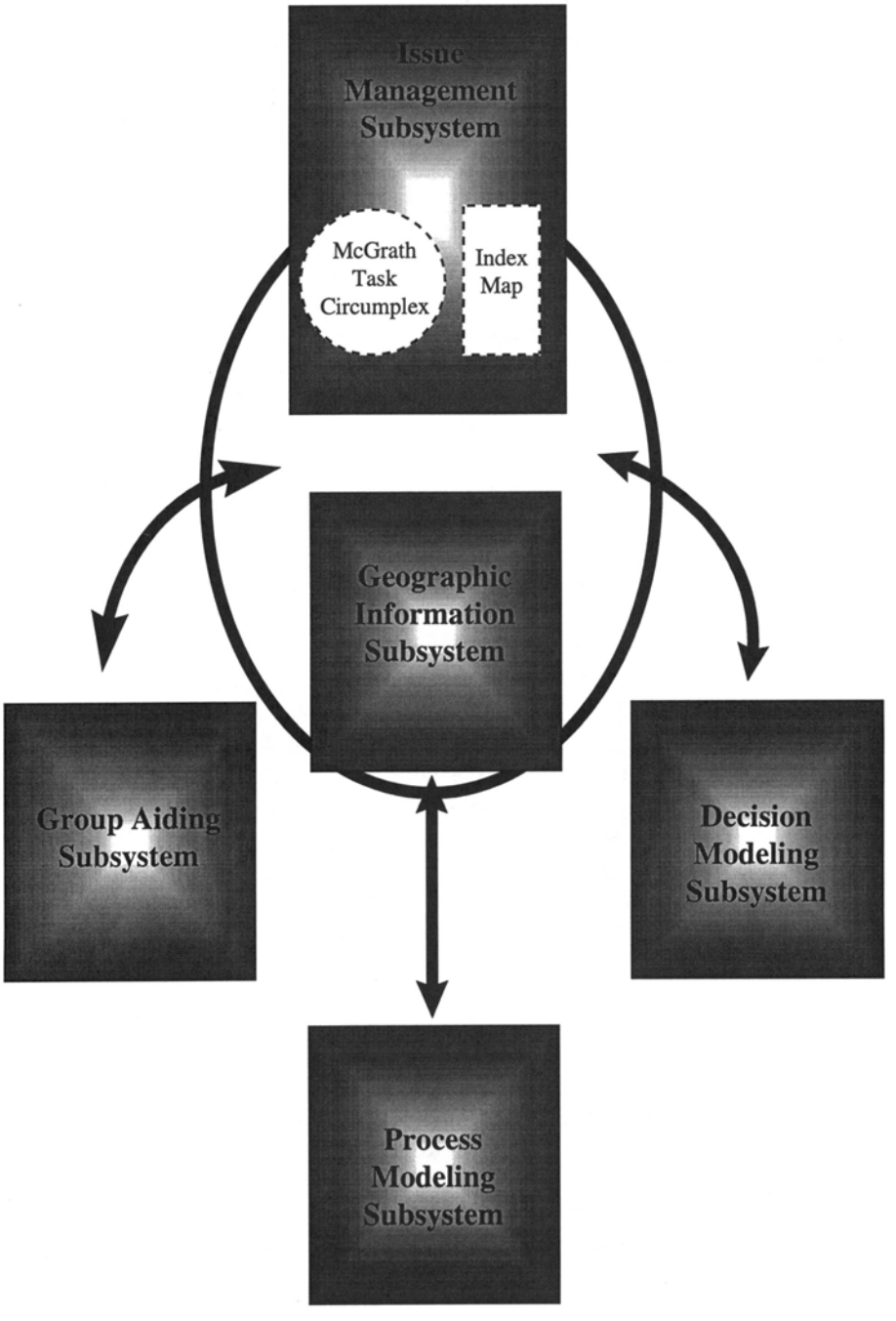


Figure 1. Proposed SUDSS Components and Schema.

going discussion (a node), the participants could create and attach to the nodes, through hyperlinks, multimedia objects which others could then view and consider. Thus, a user could see the conversation, and the "discourse artifacts" (i.e., the multimedia objects), through shared and personal contexts (geographic and otherwise).

A "Language" for an IMS

The IMS is the central organizing interface for display, modification, and retrieval of text, maps, and other multimedia data posted by the discussants in support of their computer-based discussion. In this paper, the basis for the presentation assumes the rhetorical structure proposed originally by Rittel and Webber (1973), and Kunz and Rittel (1970). That is, the IPA (for Issue, Position, Argument) structure of their proposed, and eventually tested, Issue-Based Information System (IBIS). This approach to providing some underlying conceptual organization to the nodes is based on the belief that the model for problem-solving by "cooperatives" must be viewed as an argumentative process. Other possibilities are available (such as Lee, 1990; Cropper, Eden, and Ackermann, 1993), and this could be the subject of further research.

The general structure of the IBIS method is displayed in Figure 2. Three types of nodes can be posted to the discussion. Issues raise the specific questions which participants wish to discuss. Positions propose certain approaches, or solutions, to address the issues. Discussants can then post arguments which support or oppose positions posted in the discussion. These nodes are linked by eight fundamental relationship types as depicted in Figure 2. A software, implemented non-commercially as gIBIS (for "graphical IBIS"), is described elsewhere in great detail (e.g., Conklin and Begeman, 1989). The software has led to a commercial product with friendlier node types and icons (such as "questions," with a question mark as an icon, instead of an "issue").

This proposal goes further than just a rhetorical structure. In order to provide structure within a decision-making context, the IMS interface would also prompt discussants to categorize newly created nodes by the eight task elements in McGrath's task circumplex (McGrath, 1984). This circumplex is a circle with four quadrants: Generate; Choose; Negotiate, and Execute. (In fact the quadrants are further divided in half to generate eight different decision-task slices.) This feature will impose further structure on the thinking of participants; it will require that discussants think not only about how their contributions fit into a rhetorical structure, but also how they fit within a decisional structure.

A final structure is, of course, a geographic structure. By use of an index map in the interface, each discussant can identify the approximate geographic

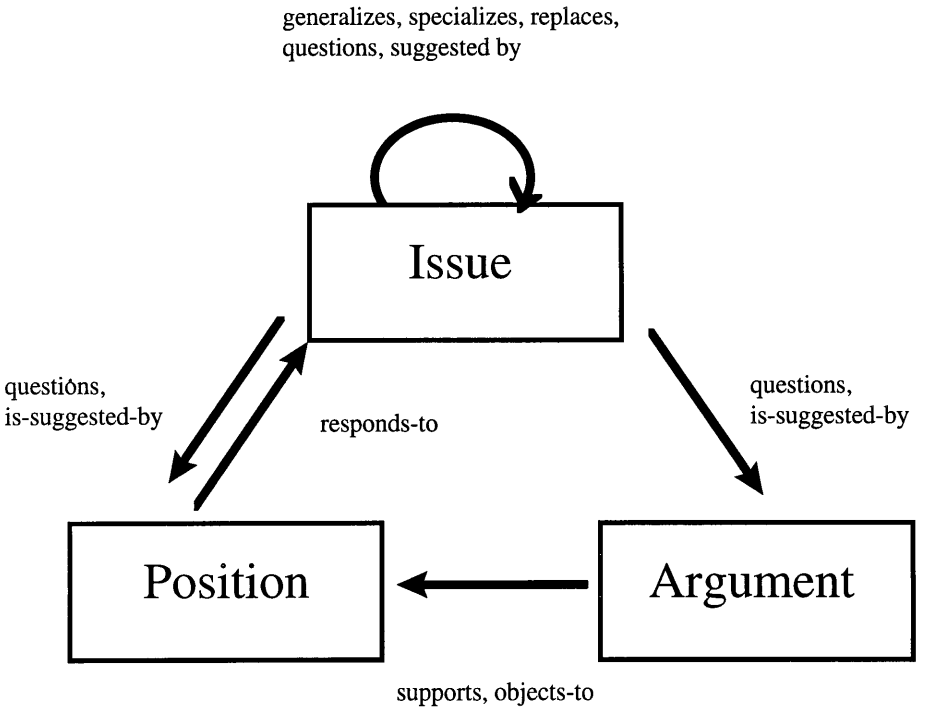


Figure 2. General Structure of the IBIS Method (as in Kim, Suh, and Whinston, 1993).

location that is related to the submitted contribution. The location can be used later to identify nodes based on spatial queries using bounding rectangles, selected features, etc. Furthermore the result of other queries (e.g., complex task/topic queries) may help to identify certain geographical patterns which could further illuminate the discussions and directions necessary for achieving consensus.

Possible Features of an IMS

Figure 3 displays a cursory example of how the IMS interface may look. Note that through pull-down menus or icons, the other subsystems could be accessed (e.g., the GIS interface, a process modeling interface, or the decision modeling or group aiding interface). The issue management interface is first divided up into topic areas (like "meeting rooms") which can be browsed, queried, and updated by discussants. Before the user clicks on a meeting room, all that can be seen is a highly generalized, or "zoomed out," view of the hypertrails that represent the various discussion topics.

Once a topic is selected, the next view shows a more detailed representation of the discourse map (the hypertrail of nodes and links). The user can navigate around this map by using zooming and panning tools, and the "location" in the discourse map is always displayed by a "bird's eye view" of the hypertrail (with a bounding rectangle to represent the view boundaries) in a corner of the map. The nodes are identified by some author-supplied keyword/title and by icons which symbolically represent the type of node. Also, the software would use a document-centered convention, so that documents created in other applications and in other subsystems can be attached to nodes in the computer-supported discussions.

Query capabilities would be available to allow the user to select and browse nodes and supporting documentation that meet specific user interests. This query tool could also be used to explore patterns in the discussions. When a query is submitted, all selected nodes will be shaded a similar color for easy visualization on the discourse maps. Additionally, if a selected node for a query has associated features on a map which were identified by a contributing author, then a minimal bounding rectangle (see Frank, 1988), or MBR, will be highlighted in the index map. If desired, the user can zoom in to the selected objects in the index map. One might envision the interface for the index map to follow upon work previously disseminated in this area (e.g., Evans, Ferreira, and Thompson, 1992).

To capture the essence of how the conversation develops over time, and to possibly assist a user in browsing along a discourse "thread," the time at which a node is added to the discourse could be used by a geographic brushing tool.

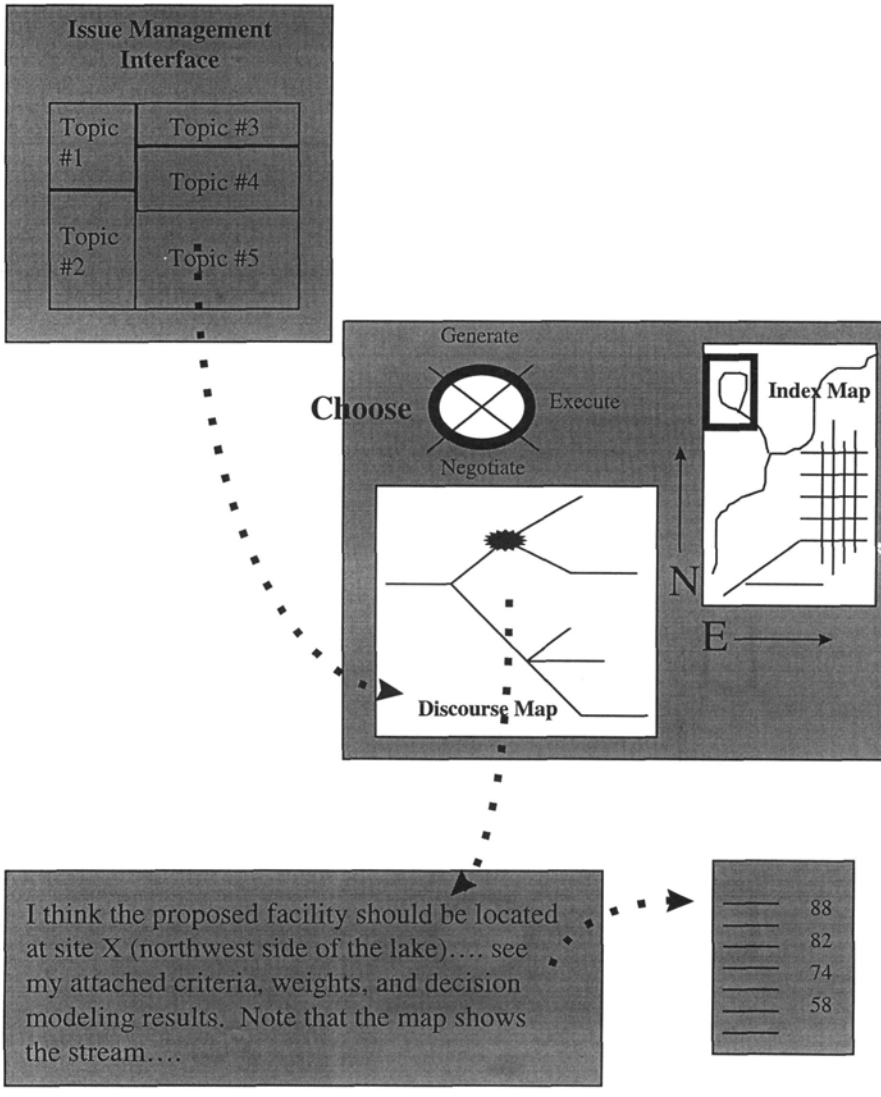


Figure 3. Hypernavigating the SUDSS Interface.

Monmonier (1992) described a research application to study the effectiveness of brushing at communicating information via dynamic maps. Here, we might imagine that as the user sweeps across a timeline, the discourse map nodes are cumulatively highlighted in the sequence in which they were contributed to the discussion. The user can stop at any point on the timeline to investigate the extent of the conversation at that selected time. Likewise, the decision task circumplex "slices" and the discussion nodes and MBRs in the index map could be cumulatively highlighted based on the sweep along the timeline.

DISCUSSION AND CONCLUSIONS

For GIS research and development, the creation of SUDSS-like tools for COMMAND presents a significant future challenge. Even in conventional settings, with non-GIS software, the software development process does not always take full advantage of usability testing methods. However, Internet-based usability studies could make the process of testing the software somewhat easier. The issues of why and how to perform these tests for collaborative spatial decision-making activities and for the development of SUDSS-like tools is beginning to receive attention (Golay and Nyerges, 1995; Nyerges, 1995; Nyerges and Jankowski, 1994, 1997; and Jankowski and Stasik, 1996).

One of the potential benefits of this approach would be that the activity to support an on-going, long-term, multiparty discussion would also create a database that can be searched at a later date. In effect, the records archive would be digitally catalogued for posterity while the policy discussions are occurring. This could improve the "institutional memory" of participating decision-making organizations. If successful, this approach could have enormous potential for long-term discussions related to policy and planning – for example, state or federal environmental review processes.

Another possible benefit is for people who cannot attend public meetings because of an odd work schedule, conflicting family responsibilities, etc. The distributed nature of a SUDSS in space and time, if Internet-based in implementation, would provide at least the opportunity for the information to be readily accessible at home. Whether, or how, the average citizen would use a SUDSS-like tool is admittedly still a question. However, the excitement about emerging technologies on the World Wide Web suggests a real and wide interest in hypermedia-based, distributed communications.

However, for each potential benefit there is also probably a cost. For example, we could easily turn the above discussion about "greater access" into a discussion about the danger of broadening the gap between the information "haves" and "have-nots." The actual (as opposed to intended) effect of a

technology in society is bound not only to its capabilities and applications, but also to the historical, political, and social milieu surrounding the tool's use.

The point that I would like to make here is that the hierarchical structure of existing GIS technology, along with the one-dimensional nature associated with its application, both tend to promote particular solutions according to specific "world-views." We need to promote further research and development into problem-solving and decision-making techniques with this technology, but we must also emphasize exploration and communication among individuals and groups with many different "world-views." I think that what we are learning is something that the influential words of Ian McHarg identified so beautifully, but it is something that, in my opinion, has never been completely captured within this technology. (McHarg, 1969) In a chapter on the health of cities, McHarg writes:

> "...the surviving and successful organism, species and community are fit for the environment. If this is so, then the fit creature or community is, by definition, creative; the unfit or misfit, less than fully creative, or reductive. As we have seen, fitness must be revealed in form, and so form then reveals not only fitness but creativity. Form is meaningful, revealing the adaptive capabilities of the organism and the community; it should also reveal, if we could observe this, that these are creative." (McHarg, 1969, pp. 187-188)

In this sense, movement toward a SUDSS design could open up the collective creativity that shapes the form of planning and public policy, for all to then contribute and view.

This, I believe, is what Daniel Sui means when he calls for a new GIS leading to a more democratic society. He says we need: ". . . a shift of our philosophy from viewing GIS as an instrument for problem-solving to viewing it as a socially embedded process for communication." (Sui, 1996) When we get to that point, then we will be able to respond to Nick Chrisman's challenge of dealing: "... with mutli-disciplinary ways of knowing, as implemented in competing GIS representations." (Chrisman, 1996)

What we are seeing in GIS research, in other areas of academia, and in society in general, is a search for ways to improve the interaction of voices and connect this interaction to real action for the common good of a community. The goals of increasing public participation and building a sense of community in a postmodern world are not easy to achieve, whether it is in the field of GIS or elsewhere. For GIS technology to improve its contributions to "societal" decision-making, we must recognize that the underlying structure of the tool must respond to three critical human needs for "community"-building: 1)

increasing communication and participation, 2) increasing connections and obligations to others, and 3) respecting the individuality of others (adapted from Daly and Cobb, 1989). So as I see it, SUDSS in COMMAND could be a very unique and valuable form of computer-mediated communication. That's because this technology may not let us soar, freeing us from the so-called "Tyranny of Geography," but rather it may just make it easier for us to put our feet back down on the ground.

References

Armstrong, M.P. (1994). Requirements for the development of GIS-based group decision support systems. *Journal of the American Society for Information Science*, 45(9): 669-677.

Carver, S.J. (1991). Spatial Decision Support Systems for Facility Location: A Combined GIS and Multicriteria Evaluation Approach. Proceedings of 2nd International Conference on Computers in Urban Planning and Urban Management, Oxford, United Kingdom, July, pp. 75-90.

Chrisman, N. (1996). GIS as social practice. Position paper, NCGIA Research Initiative 19, http://www.geo.wvu.edu/www/i19/chrisman.

Conklin, J., and M.L. Begeman (1989). gIBIS: A tool for all reasons. *Journal of the American Society for Information Science,* 40(3): 200-213.

Couclelis, H., and M. Monmonier (1995). Using SUSS to resolve NIMBY: How spatial understanding support systems can help with 'Not in My Backyard' syndrome. *Geographical Systems,* 2: 83-101.

Cropper, S., C. Eden, and F. Ackermann (1993). Exploring and negotiating collective action through computer-aided cognitive mapping. *The Environmental Professional,* 15: 176-185.

Daly, H.E., and J.B. Cobb (1989). *For the Common Good: Redirecting the Economy toward Community, the Environment, and a Sustainable Future.* Beacon Press, Boston.

Densham, P.J. (1991). Spatial decision support systems. In: D.J. Maguire, M.F. Goodchild, and D.W. Rhind (eds.), *Geographical Information Systems: Principles and Applications.* John Wiley & Sons, New York.

Evans, J.D., J. Ferreira, Jr., and P.R. Thompson (1992). A visual interface to heterogeneous spatial databases based on spatial metadata. Proceedings of the 5th International Symposium on Spatial Data Handling, Volume 1, IGU Commission on GIS, Charleston, South Carolina, August 3-7, pp. 282-293.

Faber, B.G., J. Knutson, R. Watts, W.W. Wallace, J.E. Hautaluoma, and L. Wallace (1994). A groupware-enabled GIS. Proceedings, GIS '94 Symposium, Vancouver, British Columbia, February, pp. 551-561.

Frank, A.U. (1988). Requirements for a database management system for a GIS. *Photogrammetric Engineering and Remote Sensing*, 54(11): 1557-1564.

Golay, F., and T.L. Nyerges (1995). Understanding collaborative use of GIS through social cognition. In: T.L. Nyerges, D.M. Mark, R. Laurini, and M.J.

Egenhofer (eds.), *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems.* Kluwer Academic Publishers, Boston.
Heywood, D.I., and S.J. Carver (1994). Decision Support or Idea Generation: The Role of GIS in Policy Formulation, Angewandte Geographische Informationsverarbeitung VI, Salzburg.
Jankowski, P., T.L. Nyerges, A. Smith, T.J. Moore, and E. Horvath (1997). Spatial group choice: A SDSS tool for collaborative spatial decision making. Under review with *International Journal of Geographical Information Systems.*
Jankowski, P., and M. Stasik (1996). Architecture for space and time distributed collaborative spatial decision making. GIS/LIS '96 Proceedings, American Society for Photogrammetry and Remote Sensing, Denver, Colorado, Nov. 17-21, pp. 516-526.
Kim, W., Y. Suh, and A.B. Whinston (1993). An IBIS and Object-Oriented Approach to Scientific Research Data Management. *The Journal of Systems and Software*, 23: 183-197.
Kunz, W., and H.W.J. Rittel (1970). Issues as Elements of Information Systems. Working Paper No. 131, Center for Planning and Development Research, Institute of Urban and Regional Development, University of California, Berkeley, USA, July.
Lee, J. (1990). SIBYL: A tool for managing group decision rationale. Proceedings of the Conference on Computer-Supported Cooperative Work, ACM SIGCHI & SIGOIS, Los Angeles, California, October 7-10, pp. 79-92.
Malczewski, J. (1996). A GIS-based approach to multiple criteria group decision-making. *International Journal of Geographical Information Systems*, 10(8): 955-971.
McGrath, J.E. (1984). *Groups: Interaction and Performance.* Prentice-Hall, Inc., Englewood Cliffs, NJ.
McHarg, I.L. (1969). *Design with Nature.* The Natural History Press, Garden City, New York.
Monmonier, M. (1992). Time and motion as strategic variables in the analysis and communication of correlation. Proceedings of the 5th International Symposium on Spatial Data Handling, Volume 1, IGU Commission on GIS, Charleston, South Carolina, August 3-7, pp. 72-81.
Nyerges, T.L. (1995). Cognitive task performance using a spatial decision support system for groups. In: T.L. Nyerges, D.M. Mark, R. Laurini, and M.J. Egenhofer (eds.), *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems.* Kluwer Academic Publishers, Boston.
Nyerges, T.L., and P. Jankowski (1994). Collaborative spatial decision-making using geographic information system displays and multicriteria decision-making techniques. NSF Proposal, funded 1994-1996, Division of Social and Behavioral Science, SBR-9411021.
Nyerges, T.L., and P. Jankowski (1997). Framing GIS-supported collaborative decision making: Enhanced adaptive structuration theory. Under review in *Geographical Systems.*

Paradis, J., and K. Beard (1994). Visualization of spatial data quality for the decision-maker: A data-quality filter. *URISA Journal,* 6(2): 25-34.
Rittel, H.W.J., and M.M. Webber (1973). Dilemmas in a general theory of planning. *Policy Sciences*, 4: 155-169.
Shiffer, M.J. (1992). Towards a collaborative planning system. *Environment and Planning B*, 19(6): 709-722.
Shiffer, M.J. (1995). Geographic interaction in the city planning context: Beyond the multimedia prototype. In: T.L. Nyerges, D.M. Mark, R. Laurini, and M.J. Egenhofer (eds.), *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*, Kluwer Academic Publishers, Boston.
Sui, D.Z. (1996). GIS and society: From instrumental rationality to communicative rationality. Abstract for paper yet to be published.

DATA QUALITY IMPLICATIONS OF RASTER GENERALIZATION

Howard Veregin and Robert McMaster
Department of Geography
University of Minnesota
267-19th Ave S, Rm 414
Minneapolis MN 55455

ABSTRACT

This study is concerned with the data quality implications of raster generalization. The study focuses specifically on the effects of neighborhood-based generalization (categorical filtering) on thematic accuracy. These effects are examined empirically using raster land cover maps. Accuracy is defined in terms of changes in class membership between original and generalized maps. Results indicate that changes are concentrated in those portions of the map and for those classes that exhibit high levels of spatial variability.

INTRODUCTION

Generalization in a raster environment is fundamentally different from generalization in a vector environment. In a vector environment the spatial and thematic components can be generalized independently, while in a raster environment generalization is almost always accomplished by manipulating the thematic component alone. Raster generalization changes the thematic content of maps and thus has implications for thematic accuracy and data quality in general. This study examines the effects of raster generalization on thematic accuracy for categorical data.

Raster Generalization

Several authors have developed frameworks for classifying raster generalization operators. According to the framework developed by McMaster and Monmonier (1989) the four fundamental operators are structural generalization, numerical generalization, numerical categorization and categorical generalization. Schylberg (1993) adds a set of area-feature operators which perform generalization on raster objects defined as clumps of contiguous cells with the same class.

Three classes of operators apply to categorical data. Local operators work directly on attribute values and ignore neighborhood effects. Neighborhood operators are based on class frequencies within a neighborhood or kernel. Object-based operators are applied to raster objects.

This study focuses specifically on a neighborhood operator known as modal filtering. Filtering reduces high-frequency variation in order to enhance the clarity of presentation. In "simple" modal filtering, a kernel is centered on a cell and the modal class within the kernel is determined. This modal value replaces the class of the cell at the center of the kernel. The process is repeated for every cell in the map, except those along the edges (Fig. 1). Kernel size can vary, with larger kernels producing higher levels of generalization.

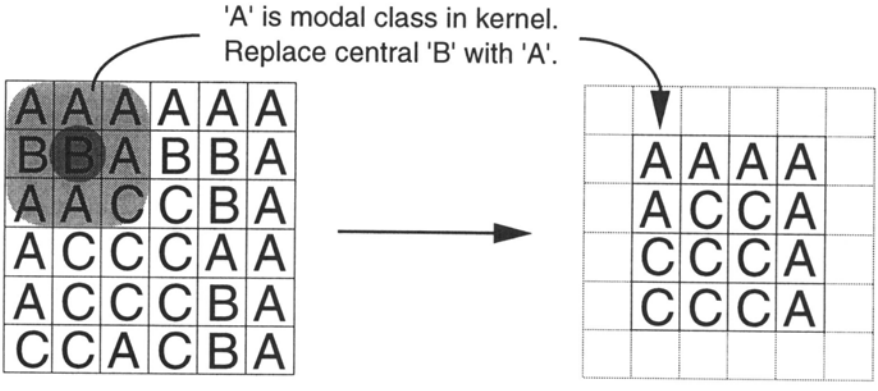


Figure 1. Simple Modal Filtering.

In this example, all classes have equal priority weights. However, unequal weighting is usually required because it is often the case that more than one class has the same frequency in the kernel. When unequal weights are employed, the modal class is defined as the class with the highest weighted frequency. There are several ways to calculate weights. They can be computed based on the area of each class over the entire map. This gives precedence to classes that cover a larger area. Alternatively, weights can be provided by the user. This is useful because it allows for selective enhancement or suppression of certain classes. Finally, weights can be computed by determining class frequencies within a neighborhood outside the kernel. This neighborhood is called a "halo" and its size can vary. A halo "bias factor" can also be defined to give the precedence of frequencies within the halo relative to frequencies within the kernel (Monmonier, 1983).

Whatever the specific method employed, filtering changes the thematic content of the original map by modifying the class memberships of certain cells. These modifications represent a form of thematic error that can be quantified using standard thematic accuracy assessment techniques.

Thematic Accuracy Assessment

Methods of thematic accuracy assessment depend on the measurement scale of the attribute under consideration. For categorical data such as land cover, the most common method is based on the confusion matrix. The matrix, denoted as **C**, has dimensions kxk, where k is the number of classes. Element c_{ij} in the matrix represents the number of cells encoded as class i that actually belong to class j. Correct classifications are those for which i=j. This occurs along the principal diagonal of the matrix. Misclassifications are those for which $i \neq j$. (For a summary of the confusion matrix as applied to classification accuracy assessment in remote sensing see Congalton, 1991).

In the case of modal filtering, an error is defined as a cell with a different class on the original and filtered maps. The confusion matrix tabulates these differences. Element c_{ij} in the matrix represents the number of cells with a class of i on the filtered map and a class of j on the original map.

The information contained in the confusion matrix is typically summarized using indices of thematic accuracy. One such index is PCC, or the proportion of cells that are correctly classified. The maximum value of PCC is 1, which occurs when there is perfect agreement. For modal filtering PCC is defined as the level of agreement between the original and filtered maps. A value close to 1 indicates that the original and filtered maps are nearly identical.

It is usual to distinguish between omission and commission errors in the classification error matrix. An omission error occurs when a cell is omitted from its actual class, i.e., a cell that actually belongs to class j is assigned instead to class i. In the classification error matrix, the off-diagonal elements that occur in a given column j are omission errors in that they represent cells that have been erroneously omitted from class j. Commission error refers to the insertion of a cell into an incorrect class, i.e., a cell is assigned to class i but actually belongs to class j. In the classification error matrix, the off-diagonal elements that occur in a given row i are commission errors in that they represent cases that have been erroneously included in class i.

Any error of omission is simultaneously an error of commission and vice versa. In modal filtering, an error is defined as a cell with a different class on the original and filtered maps. This is an omission error since the cell has been omitted from the class assigned on the original map, and a commission error since the cell has been assigned to a different class than that on the original map.

METHODS

Hypotheses

The effects of filtering on thematic accuracy are hypothesized to be non-uniform spatially and thematically. Filtering reduces high-frequency variation,

such that its effects on accuracy will be most significant in those portions of the map and for those classes that exhibit high-frequency spatial variation.

High-frequency variation is characteristic of classes that are fragmented into small, isolated patches or long, narrow ribbons. These classes tend to lack dominance at the neighborhood level and thus tend not to form the modal class within kernels. These classes will therefore tend to be suppressed by filtering, inducing errors of omission in the filtered map. These effects are reversed for classes that exhibit low-frequency spatial variation, such as those that occur as large, homogeneous clumps. These classes tend to be dominant at the neighborhood level and thus frequently form the modal class within kernels. These classes will therefore tend to be enhanced, resulting in errors of commission in the filtered map.

It is probable that these effects will be non-uniform spatially, since spatial variability is itself variable over space. Changes in class membership will tend to occur in those portions of the map in which variability is highest. These effects will also be affected by kernel size, since the degree of spatial variability is dependent on spatial scale.

Data

Data for this study were derived from aerial video imagery of the Mud Run urban watershed in Akron, Ohio. Imagery was acquired in December, 1994, using a color video camera and was post-processed to extract three spectral bands. Post-processing also included resampling to a 1-meter cell size. Supervised classification was performed using a minimum distance to means classifier (Veregin et al, 1996). The original classified map is shown in Figure 2.

Fig. 2. Original Map.

The area contains a mixture of residential and commercial buildings interspersed with transportation features, grass and bare soil. Areas of deep

shadow are common due to the low sun angle at the time of data collection. (These areas were classified as shadow rather than as their true class due to the limited amount of spectral information that could be extracted from shadow areas.) Different classes exhibit different degrees of spatial variability. For example, grass tends to occur in large homogeneous clumps, while transportation classes (especially concrete) are more linear. Other classes such as roofs and shadows occur as small, isolated clusters.

Methods

The effects of filtering were assessed by comparing the original and filtered maps. To facilitate hypothesis testing, various statistics were computed.

- Confusion Matrix. Element c_{ij} of this matrix represents the number of cells with a class of i on the filtered map and a class of j on the original map.

- Agreement. An overall index of agreement was computed as sum of the diagonal elements of the confusion matrix divided by the number of cells. This is analogous to the PCC index discussed above.

- Omission. An index of omission error was computed for each class j by dividing the diagonal element in column j of the confusion matrix by the column total for j. A higher value for the index indicates less omission error. A value approaching 0 means that almost every cell with that class on the original map has been omitted from this class on the filtered map.

- Commission. An index of commission error was computed for each class i by dividing the diagonal element in row i of the confusion matrix by the row total for i. A higher value for the index indicates less commission error. A value approaching 1 for a given class indicates that almost every cell labeled as that class on the filtered map is that same class on the original map. A value close to 0 indicates that almost every cell labeled as that class on the filtered map is in fact some other class on the original map.

- Change in Area. A simple area change index was computed for each class as the row total divided by the respective column total. A value greater than 1 indicates that the class has more cells on the filtered map than on the original. A value less than 1 indicates the opposite.

- Dissimilarity. A dissimilarity index was computed for each class. This index is a measure of local variability or "texture" for categorical data. Texture can be computed for numerical data as the variance of the cell values in the kernel (Haralick et al, 1973). For categorical data, dissimilarity is defined as the proportion of cells in the kernel that have a class that is different from the class of the cell at the center of the kernel. A higher dissimilarity value means that more variability is present. To maintain consistency in spatial scale, dissimilarity was computed using a kernel of

the same size as that used for generalization. For analysis purposes, mean dissimilarity was computed for each class.

RESULTS

Simple Modal Filtering

The first set of results apply to simple modal filtering using a 3x3 kernel (Fig. 3). Overall agreement between the original and filtered maps is 0.87. As hypothesized, differences between the original and filtered maps are associated with cells having high dissimilarity. Mean dissimilarity is 0.68 for cells that change class and only 0.22 for cells that do not change. Thus, those parts of the map that exhibit thematic error tend to be areas with high spatial variability. This effect is clearly evident in Figures 4 and 5, which show the spatial pattern of dissimilarity and the spatial pattern of error, respectively.

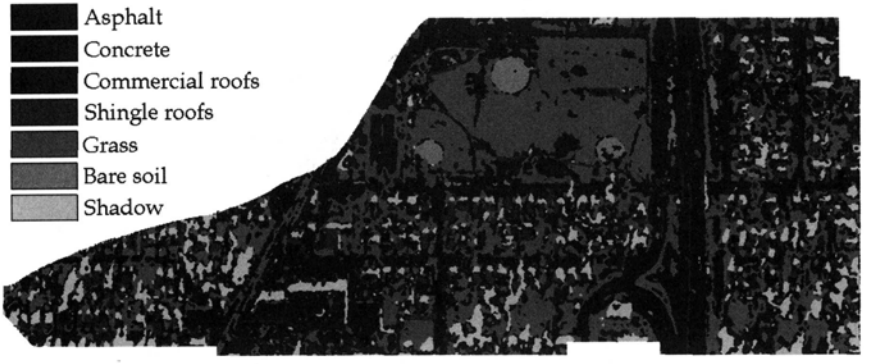


Fig. 3. Filtered Map.

Thematic accuracy statistics for each class are graphed in Figure 6 as a function of mean class dissimilarity. As this figure shows, classes exhibit different levels of dissimilarity. Classes that tend to occur in large, homogeneous clumps (such as grass) have the lowest mean dissimilarity. Classes that tend to occur as isolated patches (such as shingle roofs and commercial roofs) or in long, narrow ribbons (such as concrete) tend to have higher mean dissimilarity values.

As hypothesized, there is a thematic component associated with the effects of generalization. As shown in Figure 6, high dissimilarity is associated with a tendency for classes to be suppressed (area change < 1). Only two classes, grass and asphalt, exhibit growth (area change > 1), and both of these classes have low dissimilarity values. Bare soil and shadow appear to be anomalies, as they have low dissimilarities but tend to be suppressed by the generalization operator.

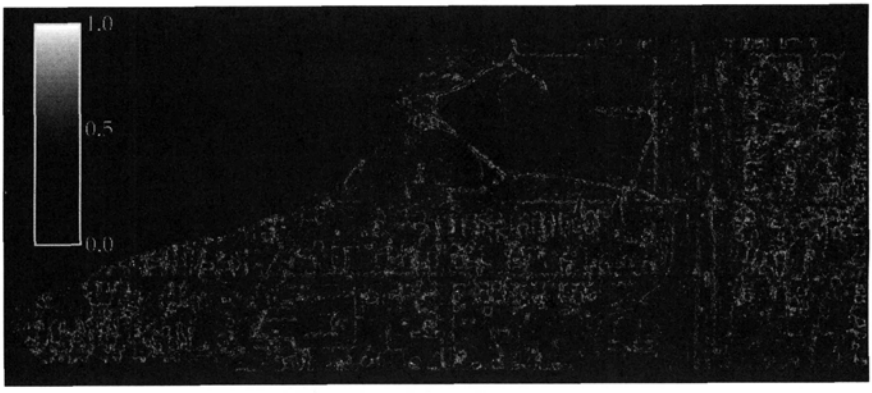


Fig. 4. Spatial Pattern of Dissimilarity Index.

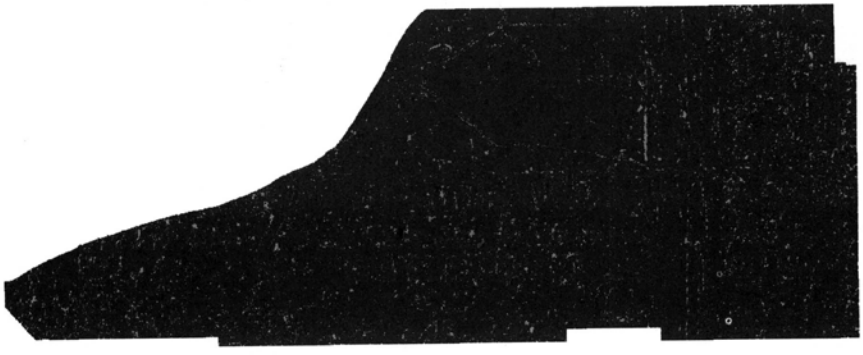


Fig. 5. Spatial Pattern of Thematic Error (Gray areas are cells with different classes on original and filtered maps).

Omission error varies across classes. Classes with high dissimilarity tend to have high levels of omission error (low omission error index). This observation reflects the fact that classes with high dissimilarity tend to be suppressed by classes with low dissimilarity, which are dominant enough to be able to form modal classes. As in the case of area change, bare soil and shadow appear to be anomalies. Figure 6 also shows that omission error and commission error are inversely related. However, there is not a clear relationship between commission error and dissimilarity.

These results support the hypothesis that filtering has the greatest impact on those portions of the map and on those classes that exhibit high-frequency spatial variation. However, mean dissimilarity seems to be an imperfect predictor of this effect. There are several reasons for this.

A high dissimilarity value for a cell means that, within the kernel centered on that cell, a large proportion of the cells are of a different class than the

center cell. However, this does not imply that these neighboring cells are all of the same class, a prerequisite for forming the modal class in the neighborhood. Thus high dissimilarity is not always correlated with a change in class membership.

- Dissimilarity may exhibit significant spatial variations that are masked by the use of mean class values. Dissimilarity for a particular class may depend on proximity to other classes. For example, bare soil might have a high level of dissimilarity when interspersed with grass, but a lower level of dissimilarity when adjacent to transportation features.

- Dissimilarity has no direct implications for commission error. High mean dissimilarity for a class indicates that the class tends to occur in proximity to other classes. This suggests a tendency for classes with high dissimilarity to be suppressed when filtering is performed. However, dissimilarity cannot be used to predict which classes will replace the suppressed classes, since it does not take into account the classes that tend to dominate in the proximity of classes with high dissimilarity values.

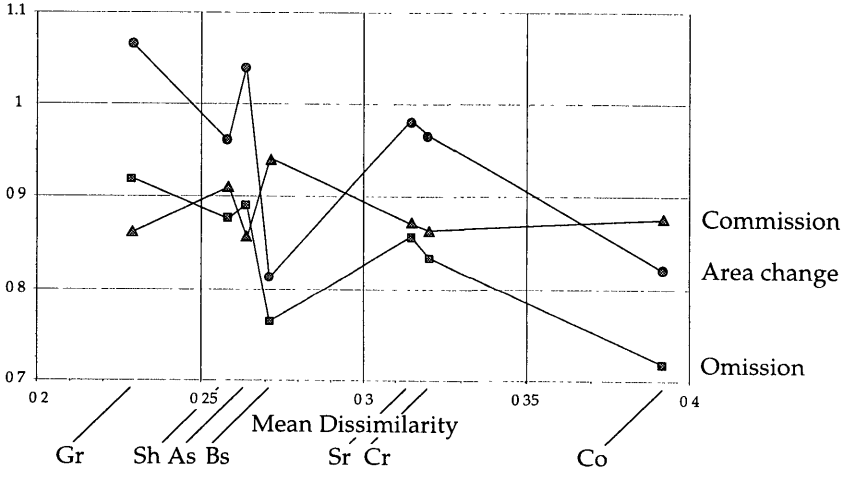


Fig. 6. Statistics for Original and Filtered Maps.

Prediction of commission error requires a measure of the tendency for different classes to exist in the vicinity of each other. One such measure is co-occurrence, which refers to the frequency with which different classes combinations occur. Co-occurrence is computed by counting the number of cells of each class within each kernel location. This yields a co-occurrence matrix, **O**, in which element o_{ij} is the number of cells with a class of i that occur within all kernels centered on cells with a class of j.

In this study, data from the co-occurrence matrix for the original map (Fig. 2) was used to predict the off-diagonal elements in the confusion matrix. The derived regression equation is as follows:

$$c_{ij} = -26.5 + 0.059\ (o_{ij} \times d_j / d_i) \qquad r^2 = 0.96$$

In this equation, c_{ij} the element in row i and column j of the confusion matrix, o_{ij} is the element in row i and column j of the co-occurrence matrix, and d_j and d_i are the mean dissimilarities for classes j and i, respectively. A large ratio of the two dissimilarity values implies that class j is more dissimilar than class i, which means that class i will tend to dominate. This implies that as the ratio increases in value, there is a greater tendency for cells of class j to be assigned a class of i on the generalized map. The high r^2 value for the regression equation indicates that class dissimilarities coupled with co-occurrence data permit reliable prediction of the off-diagonal elements of the confusion matrix.

Other Effects

Results indicate that class suppression and enhancement effects are magnified as kernel size increases. Those classes with the highest dissimilarity are all but eliminated on filtered maps when a large kernel size is used. The effects of filtering are also impacted by the selection of class weights based on the frequencies of class occurrence in a halo surrounding the kernel. Class membership is tabulated in the kernel and separately in the halo. Each of these two vectors of frequencies is then weighted by a bias factor. In this study, it was observed that the use of such weights has essentially the same effect as using a larger kernel. This is simply because the classes that tend to dominate in the halo are the same as those that dominate in large kernels.

Weights can also be defined by the user to selectively enhance or suppress certain classes. Use of these weights has a mitigating effect on the relationships between dissimilarity and thematic accuracy. In general, it is not possible to predict the effects of filtering using dissimilarity if arbitrary weights are employed. However, dissimilarity can be used to select appropriate values for these weights. High mean dissimilarity for a class implies a greater tendency for the class to be suppressed. Hence class weights that are proportional to mean class dissimilarities should ensure that classes are suppressed more evenly.

CONCLUSIONS

The results of this analysis support the hypothesis that modal filtering has the greatest impact on those classes and those parts of the original map where spatial variability is greatest. Thus thematic error introduced by filtering varies over space and theme. To our knowledge this is the first attempt to quantify the effects of raster generalization operators on thematic accuracy. Future work needs to consider the limitations of mean dissimilarity as an index of variability in an effort to enhance understanding of generalization effects and better predict

the degree of thematic error that is introduced. This would facilitate the creation of filtered maps containing low levels of thematic error and minimal omission and commission error for all classes. Future work must also consider local and object-based operators, and should focus attention on issues of visualization of generalization effects. A longer-term goal is to define rules to ascertain the types of generalization that are appropriate in different contexts in order to assure that a minimum threshold of accuracy is maintained.

ACKNOWLEDGMENTS

We wish to acknowledge the assistance of Larry Davis and Jim Jenkins in the creation of the original classified map.

REFERENCES

Congalton, R.G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment, 37*: 35-46.

Haralick, R.M., K.S. Shanmugam & I. Dinstein (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics, SMC-3*: 61-622.

McMaster, R.B. & M. Monmonier (1989). A conceptual framework for quantitative and qualitative raster-mode generalization. *GIS/LIS '89*: 390-403.

Monmonier, M. (1983). Raster-mode area generalization for land use and land cover maps. *Cartographica, 20*(4): 65-91.

Schylberg, L. (1993). *Computational Methods for Generalization of Cartographic Data in a Raster Environment.* Doctoral thesis, Royal Institute of Technology, Department of Geodesy and Photogrammetry, Stockholm, Sweden.

Veregin, H., P. Sincak, K. Gregory & L. Davis (1996). Integration of high-resolution video imagery and urban stormwater runoff modeling. *Proceedings, Fifteenth Biennial Workshop on Videography and Color Photography in Resource Assessment*, pp. 182-191.

AUTOMATIC ITERATIVE GENERALIZATION FOR LAND COVER DATA

Olli Jaakkola
Finnish Geodetic Institute
Geodeetinrinne 2 (PL 15), FIN-02431 Masala, Finland
E-mail: olli.jaakkola@fgi.fi

ABSTRACT

The problem of generalizing spatial data is studied in the context of land cover data. The theory behind automatic generalization is reviewed and a raster based deterministic iterative generalization method is proposed. The methodology is based on the Map Algebra. The multivariate statistical testing methods are extended to deal also with generalized land cover data.

The case study is related to the production of a European CORINE Land Cover map from Finland. It presents an implementation of a new methodological concept for land cover data production, supervised classification and automatic generalization.In the case study, the existing supervised classification is supported with digital maps and attribute databases. All input data is combined to a detailed land cover with a very small minimum feature size, and automatically generalized to the standard European Land Cover. According to the quality tests performed, the automatically generalized method used here meets the present quality specifications. The method is fast, and it gives an opportunity for multiple data products in various scales, optimized for different purposes.

INTRODUCTION

The study includes both a theoretical examination of generalization concepts and methodologies as well as a practical implementation of these theories to the generalization of CORINE (Coordination of the information on the environment) land cover data.

The theoretical part describes the automatic generalization as it is reviewed in present scientific papers and how much we could apply these theories to the generalization of land cover data. The theories are partly modified in order to make them fit better with the used raster-based methods. Also the operations are redefined and the guidelines for executing operations is given. The methodology for handling both spatial and semantic knowledge in generalization

operations is presented. Finally, some proposals for testing the quality of generalized land cover data are given. The accuracy testing methods are based on extensions of discrete multivariate testing, and on graphical overlays of generalized and original data.

The case study is related to the conversion of an existing land use database of Finland to a standard CORINE land cover database. At present, most of European countries still use the visual photo-interpretation of satellite images for the production of land cover features (see CORINE land cover 1992 and CORINE land cover - Guide technique 1993). In the Finnish version the conversion includes multiple input data in different spatial data forms, combination of different input data into one coverage, and a automatic generalization of the coverage to the CORINE land cover. This concept has also been tested in other European countries with common test data in Spain. The input data consisted of supervised classification, which was automatically generalized to CORINE land cover and coarser outputs. Some unlabeled areas were generalized manually. The paper shows that it is possible to automatically generalize a more detailed land cover to the coarser CORINE land cover database and fulfill the requirements specified for the database. The methodology provides a possible for multi-scale land cover databases generalized at different levels.

There is a report on the feasibility study for the Finnish CORINE land cover, with greater consideration on the theory of generalization and quality of generalized land cover data, as well as a detailed description of the contents of the case study (Jaakkola, 1994). The improved methods with heterogeneous classes are presented in Jaakkola (1995a, 1995b). The theoretical consideration is widened in Jaakkola 1996.

THE AUTOMATIC GENERALIZATION

Why, when and how to generalize

Cartographic generalization is a comprehensive process comprised of several interrelated processes. The generalization can be done for producing either graphical outputs as maps or for producing spatial data for geographical or statistical analysis. In both cases we may use similar operations, although we may not optimize the generalized data for all purposes. Therefore, it is good to keep in mind that good generalized maps are not always good spatial data for analysis. The cartographic generalization has, until so far, resisted the attempts for automatization.

Automatic generalization can be defined as the process of deriving, from a detailed (large scale) geographic data source, a less detailed (small scale) data set through the application of spatial and attribute operations. In attribute operations the spatial contiguity is not taken into account, whereas in spatial operations the value of the class at one point depends on spatial associations

with the area surrounding it.

Objectives of a generalization process are: to reduce the amount, type, or cartographic portrayal of the mapped data consistent with the chosen purpose and intended audience, or to maintain clarity of presentation at the target scale. Clearly, the objectives state that the selection of generalization operations and parameters is dependent on the intended purpose of the generalized data.

The conditions for generalization include both object based and holistic measures, which have to be specified. The object based spatial measures specify the land cover features itself, such as its size, shape or distance to other features. They can be given as parameters to the generalization operators, e.g. the minimum feature size, the minimum feature width, the minimum distance between the features, the minimum inlet/outlet width etc. The parameters used specify the content of the generalized data set. The holistic measures specify e.g. the distribution or density of the land cover features. So far, the holistic measures have not been used in the developed automatic generalization procedures.

The forms of spatial and attribute operations possibly needed for the generalization of the land cover may vary for different data sets. Presented here are the ones needed for the CORINE land cover. As the terms originally have been defined for vector transformations, in the raster domain these terms may be somewhat fuzzy (McMaster & Shea 1992). The generalization operations are combined from the standard Map Algebra (Tomlin 1990) operations. The approach given here is similar to that described as object-based raster generalization (Schylberg 1993:109).

In the generalization, we have used one attribute operation, namely reclassification, which is the regrouping of features into classes sharing identical or similar attribution. Five spatial operations have been used. Aggregation is the lassoing of a group of individual point or areal features in close proximity and representing this group as one continuous area. Merging is the joining of contiguous features together and representing it as one area. Amalgamation is the joining of contiguous feature to an another one, either by attaching the feature to the semantically closest one, or by dividing it between the neighbouring features. Smoothing is the relocating or shifting of a boundary line to plane away small perturbations and capturing only the most significant trends. Simplification is the reducing of a boundary line complexity by removing changes smaller than a certain threshold.

The operations must handle both the spatial and semantic aspects of the data. The spatial aspect is handled with the object-based spatial measures. The distance and shape measures are given directly as a single parameter for the aggregation and amalgamation. The size is iteratively increased in geometrical series until we have amalgamated the features to the specified minimum feature size. The nature of the amalgamation procedure enables to stop the

iteration at any time, and thus produce maps at all feature sizes.

The semantic information is stored in a hierarchic division of land cover classes, in a CORINE land cover nomenclature based on three levels of classes (see CORINE land cover 1992 and CORINE land cover - Guide technique 1993). In the procedures the semantic information is handled by using the groups of classes at these levels. Mainly we select groups of classes under level 1, although in merging we select classes under level 2. In the amalgamation we enlarge the possible neighbour classes from the level 1 to all land areas and to all areas, and amalgamate the features smaller than minimum feature size to nearest permitted neighbour class. In overall the generalization use hierarchy of classes specified for certain theme, at this case the theme is land cover. The method can be extended to other classifications as well by building the hierarchic groupings for the specified theme, and thus getting the priorities of classes during amalgamations.

The quality of generalized data

Most of the spatial data quality measures are scale dependent. Actually, none of the present quality measures is suitable, as such, to describe the quality of the generalized data. The generalization reduces the complexity of the data structure and adds error to the database, therefore the quality is always deteriorated in favor of simplicity and legibility. We have to understand that the error rate in the generalized database includes both the degree of generalization and the real error, the bias in summary measures and unintended positional and attribute errors produced by generalization. Also, the different generalization operations have different effects on the quality of the result (Jaakkola 1994:16-17), and the generalization operations are dependent on one another. Thus, we need to test the quality of different generalization operations combined with each other.

For testing the output data, we have modified the normal discrete multivariate analysis, based on the error or confusion matrix (see e.g. Sotkas et al. 1992), to take into account generalized land cover data. From the error matrix we have derived quality measures for the feature based attribute accuracy. Also, from the error matrix we have derived measures (Jaakkola 1994:20-21), for analysing the quality of land cover class shares after different generalization methods, namely the residual differences between the areas of the generalized and the original land cover. Visual inspection have been used for detecting the major positional and attribute errors.

CASE STUDY: THE FINNISH LAND COVER

The purpose of the Finnish Land Cover -project is to use existing land cover classification with auxiliary data and automatic generalization to produce a standard CORINE land cover database. The purpose of the CORINE land cover is twofold: to provide quantitative data on land cover for statistics, and

to provide maps of different scales for European environmental policy. The land cover database in the nominal scale of 1:100000 is considered accurate enough, but not too large in volume. The spatial measures, when to generalize, are given with a minimum mapping feature size of 25 hectares and a minimum feature width of 100 metres. The quality specifications in the original CORINE procedure include a positional accuracy limit of 75 metres, and a feature based attribute accuracy of 85 percent for a total classification (CORINE land cover - Guide technique 1993). There is no specification for the areal shares of the classes, since the original method does not include a separate generalization task.

The procedure

As an input data we propose to use existing supervised satellite data classification (forest and semi-natural areas), map masks (fields and wetlands) , statistical records with position (buildings), and statistical records with digitizing (FIGURE 1). The digitizing was not included into the feasibility study.

Firstly, we aggregated point features, namely the individual buildings to the areal feature, built-up area. Thus we expanded the ground areas of certain buildings in artificial surfaces to also cover the surrounding concrete areas. Secondly, we combined the SLAM, and the building register to an ungeneralized land cover database (FIGURE 1). At the same time with combination we reclassified 64 SLAM and some other classes to CORINE classes. Thirdly, we merged the heterogeneous classes (CORINE classes 242, 243 and 313), reclassified and statistically compared the merged features (method see Fuller & Brown 1994:10-11). Fourthly, we amalgamated small areal features to larger ones. We performed iterative amalgamations in the different hierarchic levels of the CORINE nomenclature. With the hierarchic grouping we actually defined the priorities of amalgamation, first grouping and amalgamations for each class group in the CORINE class level 1, then for all land classes in level 1, and finally for all the classes. The amalgamations were done in several iterative size levels, geometrically increasing the minimum feature size from 0,0625 hectares to 25 hectares by each time doubling the minimum feature size (FIGURE 1). Fifthly, we smoothed the border lines, and thus reduced the narrow inlets and outlets in all features with a iterative majority filtering (method see e.g. Goffredo 1995). After raster-vector conversion we simplified arcs with too many points by giving a weed-tolerance of 25 metres, and obtained the final standard CORINE land cover database from the test area.

The quality of products

The generalization level for the CORINE land cover is the area of generalized features divided by the whole area, and is about 25 percent. It could be interpreted that generalized map features have on average about one fourth of other classes included, although one third of the test area is sea, and thus in land areas the generalization level is higher.

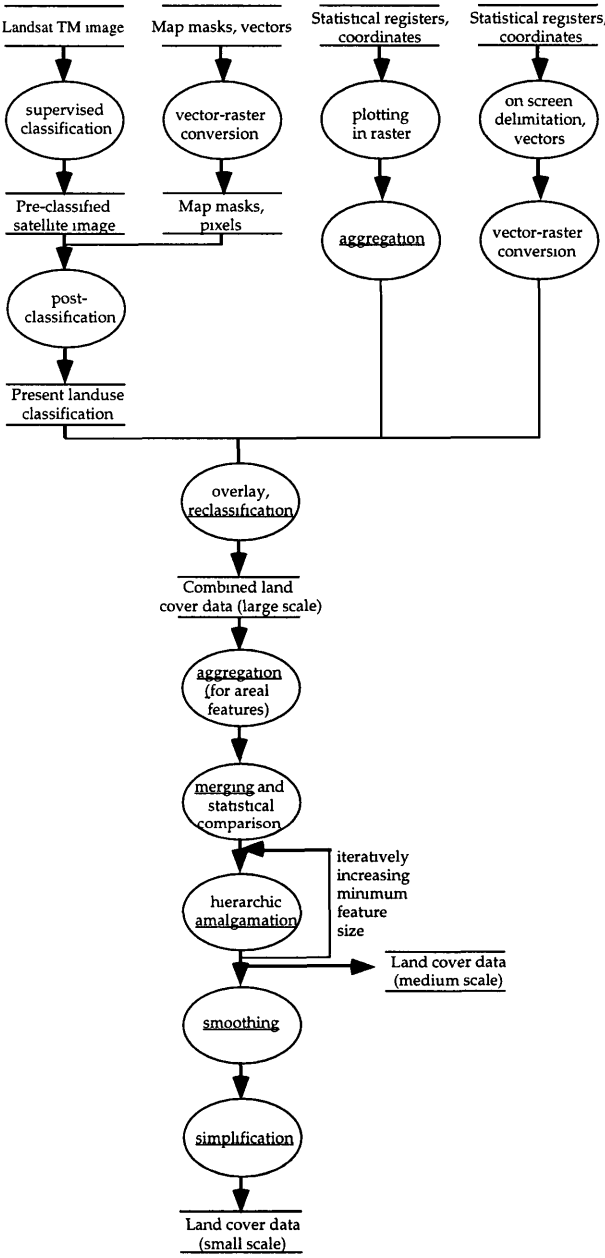


Figure 1. The classification, combination and generalization procedure.

The positional accuracy was verified by overlaying the original raster map with vector borderlines of the produced CORINE land cover. The borderlines of the main classes in level 1, e.g. water bodies, forests, and agricultural areas, are preserved quite nicely [Jaakkola 1994, appendices B5,B6]. The feature based attribute accuracy was tested with a comparison to the original combined data. In the derived measures we have the real errors produced by generalization and the level of generalization together (Jaakkola 1994, appendices A5,B5,B6]. The generalized pixels are graphically presented as a difference map between the original and generalized land cover map (Jaakkola 1994, appendix B7]. The quality of areal shares of different classes is important for the statistical use of the results. The areal shares changed systematically during the generalization, and the quality of shares were tested with the residual differences. In overall, the classes covering small areas with small average feature size tend to decrease in area or disappear totally, and the classes covering large areas with large average feature size tend to increase in area. Nevertheless, we may compensate these changes by redefining different parameters for different classes, which has not been tested yet. Note, that the residual differences (Jaakkola 1994, appendix A6) include both the degree of generalization and the real systematic errors, and that the heterogeneous classes should be reduced from the differences.

CONCLUSIONS

The presented procedures for the automatic generalization are fully operational. The processing time for the present generalization procedure for producing the standard CORINE land cover from the test area of 1200 km^2 is only a couple of hours. The automatic generalization can flexibly produce multiple products in different scales. Actually, the present procedure can produce different databases: an original combination map, maps of 1, 5, 10 and 25 hectare , or even maps of 50 or 100 hectare minimum feature size (Appendix 1). We can get all maps as side-products of the standard CORINE land cover. Maps are directly in digital form and can be stored on a database.

The accuracy tests confirm that the optimal solution fulfills the CORINE land cover quality specifications, and provides a fast, consistent, homogeneous, and accurate enough generalized land cover database. It produces a database of good positional accuracy, as well as of reasonably controlled attribute accuracy, for the class features and for the areal statistics of classes. The tests have been done in several sites in the Finland and elsewhere in the Europe. Also, the method has been tested with different minimum feature sizes for different land cover classes, and with topological constraints such as roads and other linear features. Tests has included maps as large as 40 Mega byte in size, and the method works well. The automatic generalization is considerably faster than the manual digitizing.

The size and shape parameters can be modified for different classes, e.g.

we may give a smaller minimum feature size for some important classes. Also, we can test the influence of different operations and parameters on the results, and select the most optimized ones. The procedure is areally homogeneous and objective if the supervised classification is controlled. In addition, the method is open for topological constraints, i.e. the land cover features can be kept topologically correct with certain point or linear features. The generalization method could also be used for homogenizing different data classifications in different countries, and for producing a truly consistent database in smaller scale over large areas, over the whole Europe.

Further improvement in quality can be introduced with more precise objectives. It is also good to bear in mind, that visual interpretation of the Finnish landscape is very difficult (Ahokas et al. 1992). The advantages of the new method compared to the old method are that it is fast, it provides multiple products, it provides consistent and homogeneous data, the positional accuracy is good, and the attribute accuracy is controllable since it produces systematic errors. The disadvantages is that it is not an easy task to add more detailed rules to procedure, and therefore only part of the human intelligence of interpreting is included into procedure.

Finally, we should understand that the aims of generalization can be multiple, and some procedures produce high quality maps, some high quality statistics for certain specified purpose, and these two aims can be conflicting.

REFERENCES

Ahokas, E., J. Jaakkola & P. Sotkas (1990). Interpretability of SPOT data for general mapping. European Organization for Experimental Photogrammetric Research (OEEPE) Publ. off. No. 24. 61 p.

CORINE land cover (1992). Brochure by European Environment Agency Task-Force. 22 p.

CORINE land cover - Guide technique (1993). CECA-CEE-CEEA, Bruxelles. 144 p.

Fuller, R.M. & N.J. Brown (1994). A CORINE map of Great Britain by automated means: a feasibility study. Institute of Terrestrial Ecology (Natural Environment Research Council). ITE project T02072J1. 37 p.

Goffredo, S. (1995). Knowledge-based system for automatic generalization of satellite-derived thematic maps. Proceedings of the 17th International Cartographic Conference, vol 1, pp. 108-117.

Jaakkola, O. (1994). Finnish CORINE land cover - a feasibility study of automatic generalization and data quality assessment. Reports of the Finnish Geodetic Institute, 94:4, 42 p.

Jaakkola, O. (1995a). Automatic generalization of land cover data. Proceedings of the 17th International Cartographic Conference, vol 2, pp. 1984-1989.

Jaakkola, O. (1995b). Theme-based automatic hierarchic generalization for the European Land Cover data. EUROCARTO XIII proceedings, Workshop on Scale and Extent. JRC, Ispra, Italy. pp. 80-93.

Jaakkola, O. (1996). Quality and automatic generalization of land cover data. Publications of the Finnish Geodetic Institute, n:o 122, 39 p.

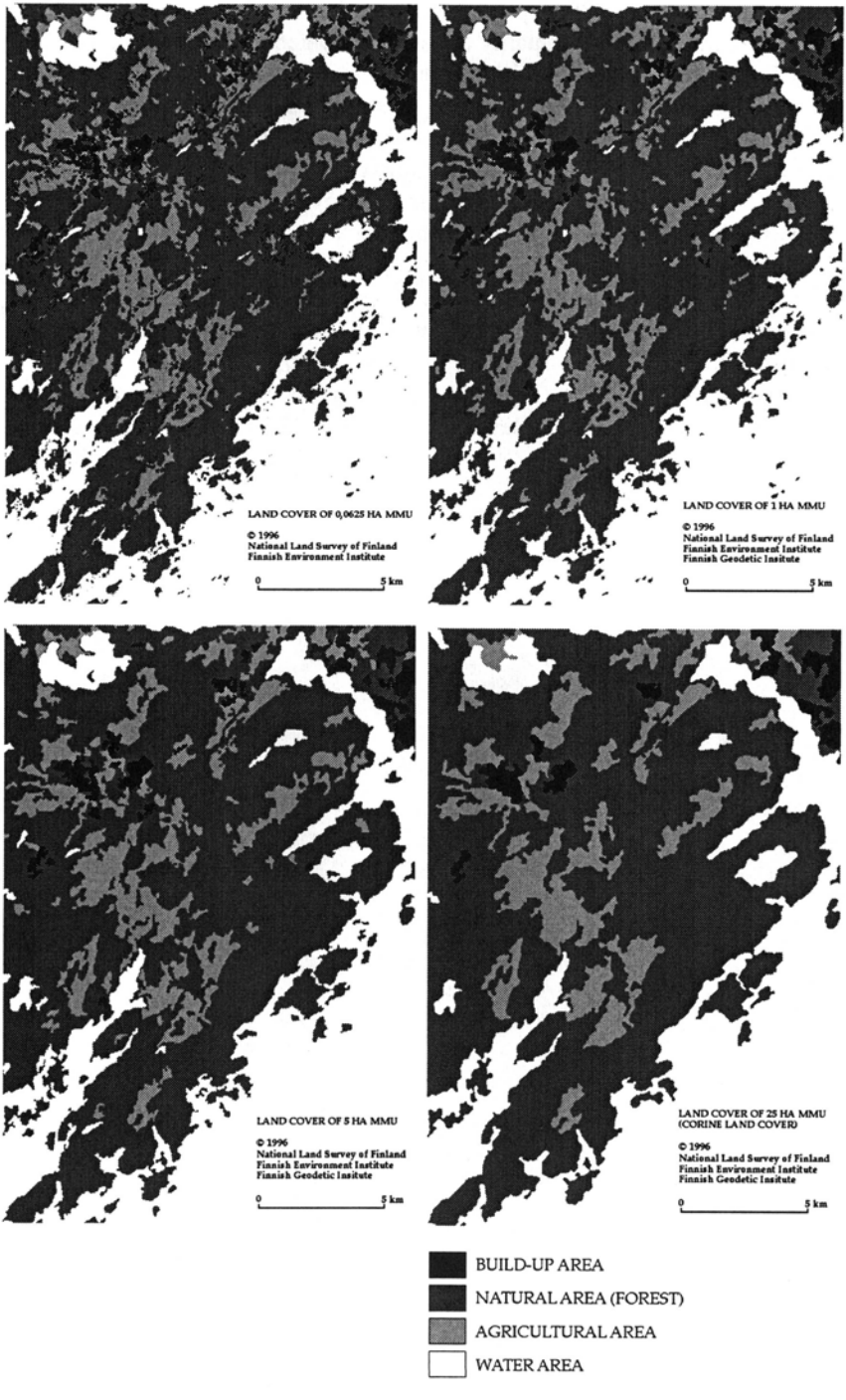
McMaster, R. & K. Shea (1992). Generalization in digital cartography. Association of American Geographers, Washington. U.S.A. 134 p.

Schylberg, L. (1993). Computational methods for generalization of cartographic data in a raster environment. Royal Institute of Technology, Department of Geodesy and Photogrammetry. Doctoral thesis. Stockholm, Sweden. 137 p.

Sotkas, P., J. Laaksonen & R. Kuittinen (1992). Satelliittikuvan tulkinta tarkkuuden määrittäminen (Determination of the accuracy of satellite image interpretations). Geodeettinen laitos, tiedote 5, 46 p. (In Finnish)

Tomlin, C.D. (1990). Geographic information systems and cartographic modeling. Prentice Hall, Englewood Cliffs, N.J., U.S.A. 249 p.

APPENDIX 1


LAND COVER OF 0,0625 HA MMU
© 1996
National Land Survey of Finland
Finnish Environment Institute
0
5 km
LAND COVER OF 1 HA MMU
© 1996
National Land Survey of Finland
Finnish Environment Institute
Finnish Geodetic Insitute
0
5 km
LAND COVER OF 5 HA MMU
© 1996
National Land Survey of Finland
Finnish Environment Institute
Finnish Geodetic Insitute
0
5 km
LAND COVER OF 25 HA MMU
(CORINE LAND COVER)
© 1996
National Land Survey of Finland
Finnish Environment Institute
Finnish Geodetic Insitute
0
5 km
BUILD-UP AREA
NATURAL AREA (FOREST)
AGRICULTURAL AREA
WATER AREA

EFFICIENT SETTLEMENT SELECTION FOR INTERACTIVE DISPLAY

Marc van Kreveld*
René van Oostrum*
Department of Computer Science
Utrecht University
email: {marc,rene}@cs.ruu.nl

Jack Snoeyink**
Department of Computer Science
University of British Columbia
email: snoeyink@cs.ubc.ca

ABSTRACT

Three new models for the settlement selection problem are discussed and compared with the existing models. The new models determine a ranking rather than a selection, which has advantages both from the efficiency and the geographic correctness point of view. We give figures of selections based on six different models, and explain how the models can be implemented efficiently.

1 INTRODUCTION

When a map is to be displayed on the screen, choices have to be made which cities and towns to include; this is called settlement or place selection (Flewelling and Egenhofer, 1993; Kadmon, 1972; Langran and Poiker, 1986; Töpfer and Pillewizer, 1966). It is intuitive that large cities should take precedence over smaller ones, but it is not true that if five cities are selected, these are the largest ones. A large city close to a yet larger city may be excluded, and a smaller city not in the neighborhood of any other larger city may be included because of its *relative importance*.

Settlement selection is performed just prior to generalization, although it can be considered as part of the generalization procedure as well. It has to be performed when a GIS user zooms out on a small scale map, or when a cartographer is in the process of interactively designing a map from a geographic database. On maps where cities fulfill a reference function, like on weather charts, clustering is undesirable, but on maps where for instance state boundaries have a reference function, clustering need not be avoided.

*Partially supported by the ESPRIT IV LTR Project No. 21957 (CGAL).
**Supported in part by grants from Canadian National Science and Engineering Research Council (NSERC), B.C. Advanced Systems Institute, and the Institute for Robotics and Intelligent Systems.

Settlement selection is followed by a name placement procedure, but we won't address that issue here. There is abundant literature on that topic.

This paper discusses models that have been described for settlement selection. We also describe a few new ones and discuss their advantages. We have implemented several of the models for comparison. In the process of interactive map design, it is useful if the cartographer has control over things like number of cities selected, and the degree in which clustering is allowed. We have included these controls in the interface of the implementation.

1.1 Previous work

Several decades ago, Töpfer and Pillewizer formalized a means to determine how many features should be retained on a map when the scale is reduced and generalization is performed (Töpfer and Pillewizer, 1966). Settlement selection itself starts by assigning an importance value to all settlements. The importance can simply be the population, but also a combination of population, industrial activities, presence of educational institutions, and so on.

Langran and Poiker report five different methods for the selection of settlements (Langran and Poiker, 1986). Most of them are incremental: cities are added from most important to least important, where the addition to the map is performed only if some spatial condition is not violated. In the *settlement-spacing ratio algorithm* and the *distribution-coefficient algorithm*, the selection of a settlement is determined by only one, more important settlement close by. In the *gravity-modeling algorithm*, selection is dependent on several settlements in the neighborhood. The *set-segmentation* and *quadrat-reduction* methods use recursive subdivision of the plane, and a direct application of the radical law by (Töpfer and Pillewizer, 1966).

Flewelling and Egenhofer discuss a number of factors that influence the selection of settlements (Flewelling and Egenhofer, 1993). Following (Mark, 1990), they assume that an importance attribute is assigned to the map features to allow for intelligent selection. Then they give a global discussion of ranking of settlements on non-spatial properties.

2 EXISTING AND NEW MODELS

Before describing the three models for settlement selection that we developed, we first discuss three existing models, reported by (Langran and Poiker, 1986): the *settlement-spacing-ratio* model, *gravity modeling* and the *distribution-coefficient-control* model. The other two methods that (Langran and Poiker, 1986) describe, *set segmentation* and *quadrat reduction*,

require too much human intervention to be suitable for automated, or interactive, map design.

A disadvantage of the three existing models is that they don't directly give a *ranking* of the base set of settlements. A ranking is a display order; after computing a ranking of the base set beforehand, selecting any number of cities is simply a matter of choosing them in order of rank. For methods that don't determine a ranking, changing the number of selected settlements involves re-computation.

Adaptations can be made to the existing models to control the number of selected settlements from the base set, but this may have strange effects. For example, when selecting more settlements, it can happen that one of the chosen settlements is no longer selected, but instead a couple of others are. When selecting even more settlements, these discarded settlements can reappear. We say that a settlement-selection model has the *monotonicity property* if any selection of settlements necessarily includes the settlements of any smaller selection. Since our new selection models are based on a complete ranking of the settlements, they have the monotonicity property.

Although ranking facilitates the selection process, a model that produces a complete ranking is not necessarily better than a model that doesn't. The quality of a final selection depends on the data set used and the purpose of the resulting map. The quality of the existing models and our new ones can be assessed by comparing figures of selections (section 4) and by our implementation available on the World Wide Web.

In the models to be described next, we assume that an importance value is known for each settlement. The model defines which settlements are selected when their geographic location is incorporated as well.

2.1 Existing Models

Settlement-spacing Ratio In the settlement-spacing-ratio model, a circle around each settlement is placed around each settlement whose size is inversely proportional to its importance. More precisely, the radius is c/i where i is the importance and c is some constant (the same for all settlements). Settlements are added in order of importance, starting with the most important one. A settlement is only accepted if its circle contains none of the previously accepted settlements. In other words: small settlements will only be accepted if they are isolated.

The constant of proportionality c determines how many settlements are accepted; smaller values for c mean smaller circles and this generally leads to more settlements being selected for display. This is, however, not always the case, as is illustrated in Figure 1: settlements 1 is accepted, 2 is rejected, and 3 and 4 are accepted. But if c were slightly smaller, the circle

of 2 would not contain settlement 1 anymore. So settlements 1 and 2 are accepted, but 3 and 4 are rejected, since their circles contain settlement 2. If we continue to decrease c, settlements 3 and 4 will reappear.

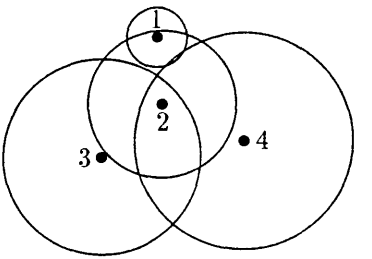


Figure 1: Settlement-spacing Ratio method

It follows that this method doesn't have the monotonicity property and that a complete ranking of the cities cannot be calculated. In fact, it can be that no value of c gives a desired number of settlements. It is also possible that two different selections have the same size. This is all caused by the fact the the monotonicity property is not respected by the model.

Gravity Modeling In the *gravity modeling* method, a notion of *influence* is introduced: the influence of one settlement on another one is computed by dividing the importance of the first one (the selected one) by the distance to the other. Settlements are tested in decreasing order of importance, and a settlement s is only accepted if its importance is greater than the summed influence of all already selected settlements on s.

As in the previous model, a constant of proportionality c is used. The next settlement under consideration is accepted if the summed influence of the already accepted settlements on the candidate is less than c times the importance of the candidate. By controlling c, the number of selected settlements can be adjusted. However, this model, like the previous one, doesn't respect the monotonicity property and doesn't give a complete ranking.

Distribution-coefficient Control The third method, *distribution-coefficient control*, uses the *nearest neighbor index* for the selection process. The nearest neighbor index is the ratio of the actual mean distance to the nearest neighbor and the expected mean distance to the nearest neighbor. Again, settlements are processed in decreasing order of importance. Starting with a small set of largest ones, settlements are only accepted if their addition to the already accepted set doesn't decrease the nearest neighbor index. The number of settlements in the final selection is fixed, but can be controlled by introducing a tuning factor. Again, this method doesn't result in a complete ranking of the settlements. A second disadvantage of the model is that the actual importance of a settlement is only used in the

order of processing, not in the selection.

2.2 New Models

Circle Growth In the *circle-growth* method, a ranking of the settlements is determined as follows: for each settlement a circle is drawn with an area that is proportional to the importance of the settlement. The initial constant of proportionality c is such that no two circles overlap. The next step is to increase c, causing all circles to grow, until the circle of some settlement fully covers the circle of some other one. The former is said to *dominate* the latter; the latter has the lowest rank of all settlements and is removed. This process is repeated while assigning higher and higher ranks, until only the most important settlement remains.

This method satisfies two important conditions:

- When two settlements compete for space on the map, the most important one of the two will survive.
- Settlements of low importance will be displayed on the map if there are no settlements of higher importance in their proximity.

The drawback of this method is that a settlement with very high importance can have a global effect on the map: its neighborhood is a large part of the map, and too many settlements near to it are suppressed. At the same time, in large regions with no settlement of high importance several settlements are selected. One way of resolving this is instead of giving each settlement a circle with an area proportional to its importance i, letting the size of the circle be proportional to i^α, with $0 \leq \alpha \leq 1$. By tuning α the influence of the importance of the settlements on the selection can be reduced.

Circle Growth Variation I The drawback of the (unmodified) circle-growth model led to the observation that settlements with a very high importance have too much influence on the selection, and this resulted in the opposite of preserving density locally. Our second method, a variation on the circle-growth method, doesn't have this problem. We'll rank from first to last this time, and give all ranked settlements a circle of the same size, i.e., proportional to the importance of the most important settlement. All not yet ranked settlements have a circle with a size proportional to their importance.

The complete ranking is calculated as follows: the settlement with the highest importance is assigned the highest rank. Next, the settlement that is second in rank is determined by applying the circle-growth idea. We choose the settlement whose circle is covered last by the circle of the

settlement with the highest rank, and set its importance to that of its dominating settlement. This process is iterated, ranking a next settlement when its circle is covered last by any one of the ranked settlements.

With this method the distribution of the selected settlements can be expected to be more even than the distribution of the selection resulting from the circle-growth method, since in our second method the size of the circles is the same for all selected settlements. Indeed, our implementation verifies this; an evenly distributed selection is the result.

Circle Growth Variation II In the previous two methods, all calculations are done with absolute importance values of the settlements. Our third method makes qualitative rather than quantitative comparisons. First, the settlements are sorted by importance from low to high. Each settlement receives as a number the position in the sorted order. This number replaces the importance value, after which the ranking is computed as before. Circles of selected settlements have equal size, and the size of the circles of the not selected settlements is proportional to their position in the list sorted on importance.

3 IMPLEMENTATION

We are currently in the process of implementing the three existing and the three new models. A preliminary stand-alone version is up and running; a java-script version is accessible via the World Wide Web at `http://www.cs.ruu.nl/~rene/settlement/`.

3.1 User Interface

The user interface will look like depicted in Figure 2: a large portion of the screen is reserved for displaying the settlements. Next to the display area are the controls: buttons for selecting which of the six implemented methods to use, or simply ranked by importance; buttons for displaying names and importance values with the settlements; buttons set the number of displayed settlements; buttons for increasing and decreasing the influence of the importance values on the selection; and a slider for adjusting the tuning factor used in the three existing models (see section 2.1).

3.2 Algorithms and Data Structures

In the first version of the program we didn't pay much attention to the efficiency of the algorithm, since our focus was on the outcome of the models rather than on speed. All models were implemented with a simple $O(n^2)$ time algorithm, or worse. However, in a practical situation like interactive map design, where data sets are large, efficiency becomes important.

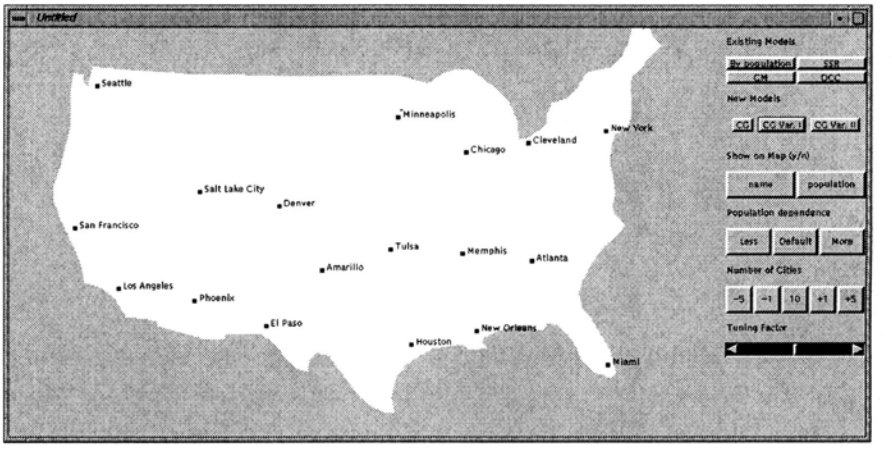


Figure 2: The user interface

Computational geometry techniques can be used to improve performance.

We'll concentrate on the implementation of the two variations of the circle-growth models, since they seem to give the best selection results. Both can be implemented by maintaining the Voronoi diagram of the selected settlements (see Figure 3). We start with one selected settlement, the most important one. Its Voronoi cell is the whole plane. Since the circles of all selected settlements have the same size, all non-selected settlements are dominated by their nearest selected neighbor. That is, their circle will be covered first by the circle of the nearest chosen settlement. So during the algorithm, we maintain for each Voronoi cell a list of non-selected settlements that lie in that cell. One of the settlements in each list is the last to be covered, and it is a candidate for the next settlement to be chosen. We maintain all these candidate settlements in a heap, which makes it possible to determine in $O(1)$ time the next settlement to be added to the set of selected settlements. Then we have to update the Voronoi diagram: a new cell is created, and a number of existing cells need to be changed. Also, the lists of non-selected settlements of the inflicted cells have to be updated, as well as the heap.

If the settlements are inserted in random order, the algorithm runs in $O(n \log n)$ expected time. In typical cases, the order in which the settlements are inserted will probably be sufficiently random for the running time to be closer to $O(n \log n)$ than to the $O(n^2)$ worst-case time.

Our first method, the unmodified circle-growth method, can be implemented in much the same way, but since here the circles of the selected settlements don't have the same size, we need weighted Voronoi diagrams

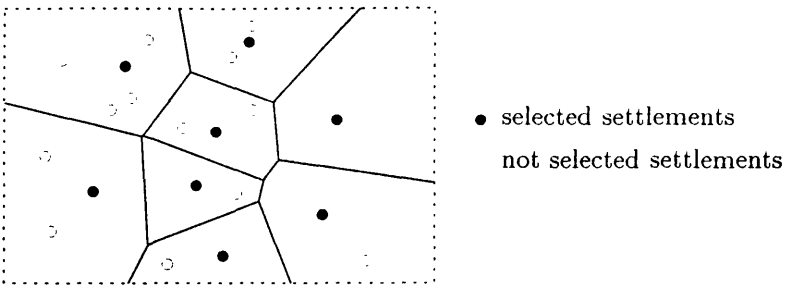


Figure 3: Maintaining the Voronoi diagram of the selected settlements

(Aurenhammer and Edelsbrunner, 1984; Okabe *et al.*, 1992), a variation of the standard Voronoi diagram that uses a different metric.

Of the existing methods, the settlement-spacing ratio method can also be implemented by incrementally constructing the Voronoi diagram of the selected settlements; a settlement is only accepted if its circle does not contain its nearest neighbor. Since settlements are added in order of importance, we don't need to maintain lists of non-selected settlements for each Voronoi cell. Note that only one complete selection is computed in $O(n \log n)$ time, not a complete ranking. So if more settlements are needed, the algorithm has to be started all over with a different constant of proportionality. In our algorithms a complete ranking of the settlements is computed in asymptotically the same time; after that, determining a selection takes time proportional to the number of settlements to be selected.

For the gravity modeling method, computing even one selection takes $O(n^2)$ time. It is not clear how to improve the performance of this model.

In the distribution-coefficient control method, testing each settlement involves determining its nearest neighbor, and determining for which of the already selected settlements the new settlement becomes the new nearest neighbor. With a straightforward algorithm this will take $O(n^2)$ time in total, but this can be improved to $O(n \log n)$ time for typical cases by incrementally constructing the Delaunay triangulation; the techniques are analogous to those for the circle-growth method. Again, this is the time needed for computing a single selection.

4 TEST RESULTS

We tested the three existing and the three new models on a (somewhat outdated) data set consisting of 158 cities of the USA. The population of the cities was used as the importance, and in each model 15 cities were displayed

(see Figure 4). For the existing models, this involved using a tuning factor. Observe that of the existing models, the gravity model and the distribution-coefficient control method show some clustering. Our unmodified circle-growth algorithm also doesn't perform very good in that respect, but the to variations result in a well-distributed set of settlements.

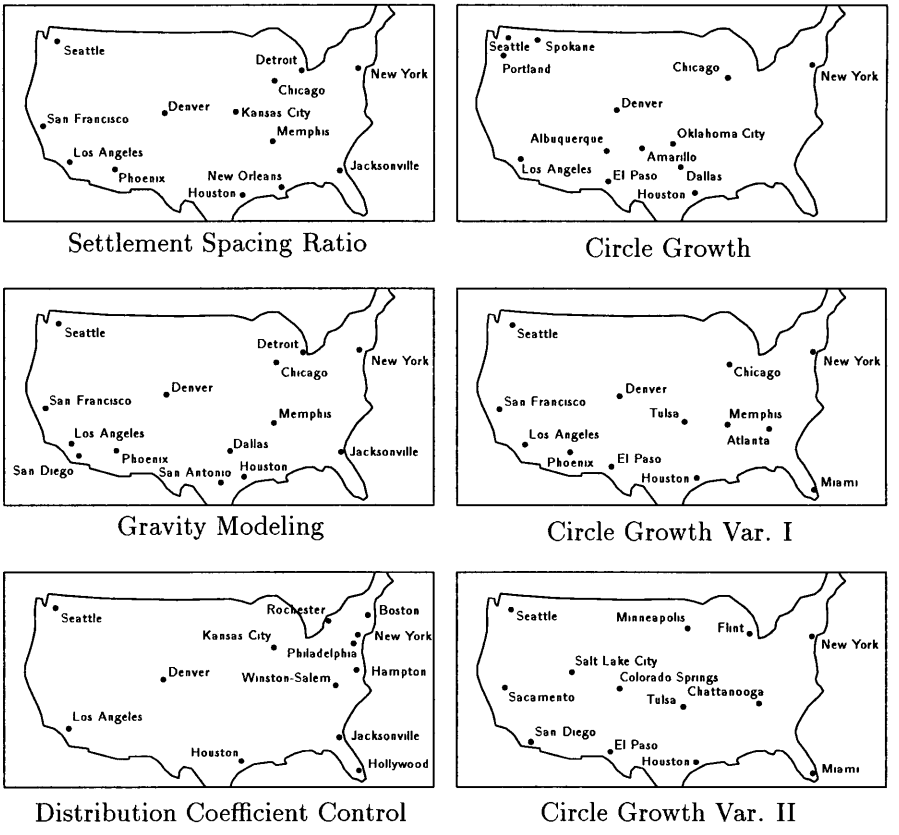


Figure 4: Results of the different methods

5 CONCLUSIONS AND FURTHER RESEARCH

We developed three new models for the settlement-selection problem and compared them with existing models. While the existing models compute a single selection, the new models determine a complete ranking of the settlements. After ranking, selecting any number of settlements is easy. Moreover, when selecting more settlements, all previously selected settlements remain selected, which is not the case in the existing models. The new methods allow efficient implementations, and result in evenly distributed

selections compared to the existing models. Adjustments to the model can be made to select the most important settlements primarily, or make a very evenly distributed selection of the important settlements.

We are planning to investigate some more variations on the circle growth model, and to come up with better ways of fine-tuning the importance dependency of the various models. Another aspect we want to look into is the effects of panning and zooming on the selection. It would also be interesting to develop methods for selection of map features that are not represented by points, such as roads, lakes, and rivers.

REFERENCES

Aurenhammer, F. and Edelsbrunner, H. (1984). An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recogn.*, 17:251–257.

Flewelling, D.M. and Egenhofer, M.J. (1993). Formalizing importance: parameters for settlement selection in a geographic database. In *Proc. Auto-Carto*, pages 167–175.

Kadmon, N. (1972). Automated selection of settlements in map generation. *The Cartographic Journal*, 9:93–98.

Langran, G.E. and Poiker, T.K. (1986). Integration of name selection and name placement. In *Proc. 2nd Int. Symp. on Spatial Data Handling*, pages 50–64.

Mark, D.M. (1990). Competition for map space as a paradigm for automated map design. In *Proc. GIS/LIS'90*, pages 97–106.

Okabe, A., Boots, B., and Sugihara, K. (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, Chichester, England.

Töpfer, F.T. and Pillewizer, W. (1966). The principles of selection. *The Cartographic Journal*, 3:10–16.

EXPLORATORY ACCESS TO DIGITAL GEOGRAPHIC LIBRARIES[†]

Vincent F. Schenkelaars
TNO Physics and Electronics Lab
Oude Waalsdorperweg 63, NL-2597 AK The Hague, The Netherlands
Schenkelaars@fel.tno.nl

Max J. Egenhofer
National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
Department of Computer Science
5711 Boardman Hall, University of Maine, Orono, ME 04469-5711, USA
max@spatial.maine.edu

ABSTRACT

Users of digital geographic libraries face the challenge of discovering the dataset they are interested in. When locating and selecting spatial datasets, different scenarios may occur with varying levels of knowledge about the datasets desired. For example, some users may have detailed information about what they want ("I need the most recent Spot image covering Orono, Maine"), while others may be more vague in their descriptions ("To test my hydrological runoff model, I need a dataset that includes a terrain with steep slopes as well as some flat areas with sandy clay"). Usually, the collection of datasets available is by far too large to be examined one-by-one. The sheer size of geographic libraries poses a performance problem for the digital library – how to retrieve enough data within a short time so that users can make decisions – as well as a cognitive overload for the users – how to select from among all datasets available, those datasets that are worth a more detailed examination. We propose an interactive geographic browser with which users can explore a geographic library by examining query results. The browser, based on the magnifying glass metaphor, allows users to move a filter over datasets displayed against a background map, while on the fly changing non-spatial parameters that determine what datasets will be visible in the magnifying glass.

[†] This research was partially supported by Intergraph Corporation and TNO-FEL under AIO-contract number 93/399/FEL. Max Egenhofer's work is further funded by the National Science Foundation under grant numbers SBR-8810917, IRI-9309230, and IRI-9411330; and grants from Space Imaging Inc., Environmental Systems Research Institute, and the Scientific Division of the North Atlantic Treaty Organization.

INTRODUCTION

The number of digital geographic datasets available either through the Internet or on CD-ROMs is increasing dramatically. The critical aspect is not anymore to find smart ways to process data efficiently, but to determine which data are relevant and find what datasets are available for the task. This is not finding the famous needle in the haystack, but finding a straw with some desired properties. Similar to the process of checking sequentially one straw after another, it would be impractical to ask users to retrieve the geographic datasets one after another until a dataset matches their expectations. Such access would be tedious and expensive – both in terms of time and in money, particularly if charging for access was based on the number of datasets retrieved. To locate the "right" spatial dataset for a project, a digital spatial library has to offer some *access tools* tailored towards the working habits of its users.

Querying a digital geographic library for the purpose of retrieving a dataset is not much different from querying a geographic information system; therefore, querying in digital spatial libraries is well supported by spatial query languages, where SQL dialects are the most prominent ones (Egenhofer 1992). If one assumes that every user of a digital geographic library will have detailed knowledge about the particular dataset he or she is interested in, access becomes a mere query that results is the desired dataset. No further searching is necessary. The important aspect with this access method is the size of the result – a small enough answer, which can be presented to the user so that he or she can understand it and exploit it without further questions. Such a scenario may fit the behavior of users retrieving a particular dataset they had accessed before and whose key characteristics they remembered, such as the type of the dataset, its geographic location, and its recording time. More frequently, however, is the scenario when users lack such specific and precise knowledge about the datasets they want to find and their query will result in a fairly large subset of the entire library. The subsequent interaction with the query result is the challenging effort, because now the user has to choose those datasets that are more promising to his or her task than the others. This is the moment when users *explore* what is available in the library, and feedback they get from the datasets may trigger new demands, or make them decide to drop a particular line of thought in their search.

We distinguish three types of interactions with a geographic library:

- *spatial querying* (Egenhofer and Herring 1993) allows users to get answers to particular questions provided they have enough knowledge about the target objects. In addition, users need some knowledge about the way the data are structured in the database.

- *Spatial browsing* (Clementini *et al.* 1990), on the other hand, is going through the answers of a spatial query and finding the interesting items. This can be considered as a human executed query process.

- *Exploratory spatial acces* (Egenhofer and Richards 1993) can be considered as a special type of browsing, in which users do not know in advance what they are looking for. Users go out into the unknown and comes back with whatever they think is interesting or appropriate for their tasks. Exploratory access is far more interactive than browsing through a query result. It is this demand for interactivity that creates the need of a special tool for exploratory browsing.

When concerned with finding datasets, exploration is most often the users' choice. A variety of implementations can be envisioned for exploratory access to spatial datasets.

- One could present the users with a (prioritized) list of resulting datasets together with some of their key properties. By scrolling through this list, users examine the characteristics and if close to their intent, they download the dataset. While such a prioritized list may lead to a good hit list if non-spatial properties are irrelevant, it lacks a connection of the datasets' spatial properties.

- To provide users some idea of the content of the datasets, small representative subsets may be used to assess spatial properties such as dispersion, density, and pattern (Flewelling 1997). Still users need supporting tools to evaluate subsets.

- Another method that exploits spatial properties of the datasets such as location and extend is the display of the datasets' outlines over the background of a reference map or an aerial photograph. Users can make spatial choices by directly examining extend and location.

This paper focuses on the latter scenario and describes a browsing method that is based on displaying spatially the datasets and allows users to filter interactively those datasets that match some non-spatial criteria. It stresses the user interface aspects of such a method and its primary concern is how to deal with the visual clutter that may result from the many overlapping and nested geographic datasets. Investigations of such a browsing tool are related to the Alexandria project, which focuses on the development of a Digital Spatial Library (Smith 1996).

The remainder of this paper is structured as follows: The next section describes an example scenario of a users task. Section 3 describes EAGLE's user interface and desired functionality. Section 4 deals with architectural aspects of the intended system. Section 5 shows an example user session of the system. Finally, section 6 discusses future research and implementation aspects.

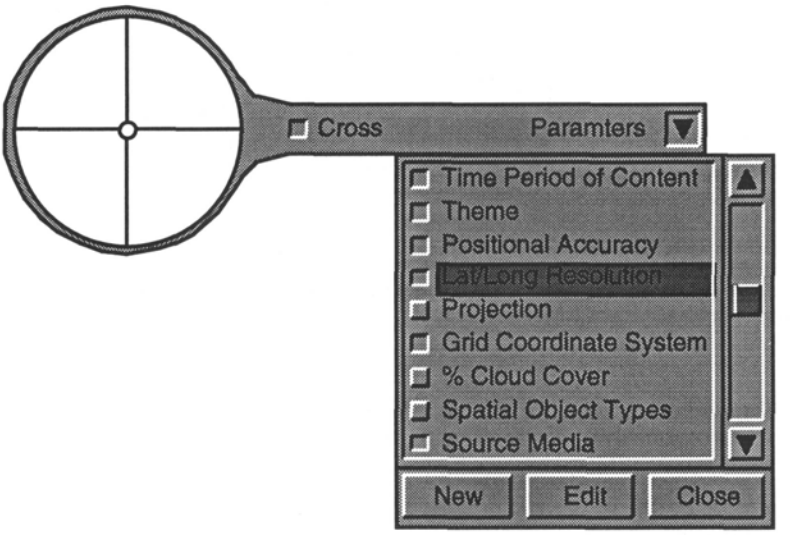


Figure 1: The magnifying glass with its list of parameters.

EXAMPLE SCENARIO

Consider a user who wants to find a dataset that can be used in some scientific project. The user has some knowledge about what the data must contain. First, this user tries to define a query and process it on a database containing the metadata information of the datasets. After execution of the query, the user is likely to end up with a very large number of datasets that match the initial requirements. After examining the metadata of a small number of the resulting datasets, the user realizes that there is another constraint that can be put into the query. So the query process has to be started again. This iterative process of adjusting a query and processing the query again is often inefficient. We argue that when the user has a tool that gives him or her direct and dynamic control over what datasets will be selected, the task of getting an appropriate dataset can be accomplished more efficiently. For this purpose, we use the magnifying glass metaphor and extend it with having filter parameters inside it. This is a combination of the movable filter (Stone *et al.* 1994) and dynamic query filtering (Ahlberg and Shneiderman 1994) concepts.

THE MAGNIFYING GLASS

The magnifying glass metaphor is the central object in the user interface. One of the important properties of the magnifying glass is that it restricts the operation area. This will dramatically improve the efficiency of the selection process. With the magnifying glass, the local operation area is moved over a

larger "overview" map, such that the user can explore a smaller area with more detail.

For the exploratory access to a geographic library, we also need a kind of filtering mechanism that will show only the qualifying target objects. This filtering mechanism needs to be very flexible. It should be possible to change the filtering parameters in an instant. Those filtering parameters are conceptually connected to the magnifying glass and we therefore connect them physically with the magnifying glass (Figure 1). The magnifying glass contains a button to switch a crosshair on and off. The crosshair may be used to focus the magnifying glass on a specific object. When the user moves the crosshair over an object on the map, the parameter values of that object are shown in a separate *attribute window*, which reflects the magnifying glass parameters, so the user can compare the actual object attribute values with these parameters. This window remains visible until the user decides to close it. This way, a user can compare the parameter values of a number of objects.

All the parameters in the magnifying glass parameter list are combined with a conjunction (logical `and`). There is little use for an `or` combination of parameters, because in that case the user is better of with an SQL-like query language. The magnifying glass is meant to be an exploratory browsing tool, so it does not need all the expression power of a spatial query language.

There is also a button on the handle of the magnifying glass which, when pressed, will show a list with possible filtering parameters. Each parameter has a check box with which it can be turned on and off. A turned-off parameter will not be considered in the filtering process. Attached to the list are three buttons: New to add a new parameter; Edit to change the values of the search parameters; and Close to close the parameter list. The New button activates a parameter creation dialog. In this dialog, the user can enter a name for the parameter, a type of parameter, and, depending on the type of parameter, a value range.

We distinguish four of different parameter types: Tri-State, Threshold, Interval, and Enumeration. These parameter types are discussed in the next subsections.

Tri-State set

This type of parameter is used to group a number of related parameters (Figure 2). A geographic dataset contains a number of spatial object types. A user might not be interested in datasets containing, for example, uniform B-splines, or the user might only be interested in datasets that contain at least point, line, and area features. Another possibility is that a user does not care if a dataset contains a certain object type. Initially, all the elements of the parameter set are in the *don't-care* state. The user moves the element to one of the other states by selecting it and dragging it via direct manipulation to the desired column.

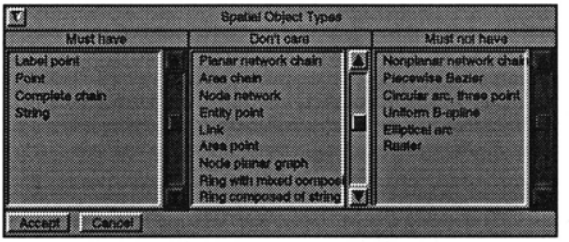


Figure 2: The Tri-State Set to select properties of objects.

Threshold

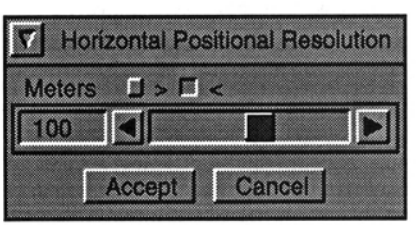
This type of parameter defines a threshold value. The target objects must have a parameter value that is larger or smaller than the threshold value. The dialog box in figure 3 shows an example of this type of parameter. The units, and ranges (maximum and minimum values) of the slider are defined when the search parameter is created.

Interval

This type of parameter defines the interval in which the parameter value is allowed. An example is a date value. A user can require that, for example the time period described by the geographic data lies between January 1 1990 and December 1 1993. The begin and end value of an interval parameter are set by moving two sliders. It is also possible to put some constraint on the sliders, e.g., end date is later than begin date (Sleezer 1994). Again, the units and ranges (maximum and minimum values) of the sliders are defined when the search parameter is created.

Enumeration Subset

This parameter type allows users to select a set of discrete values from a small set of choices. The user can select which subset of the initial set is allowed. The dialog box in Figure 4 shows an example of the data media on which the geographic dataset can be purchased. In this example the user can only use the Internet, a CD-ROM, a floppy disk, or a 9-track tape. Only geographic datasets that are available in one of those media are selected in the magnifying glass.



Figuur 3: Dialog box to select a threshold

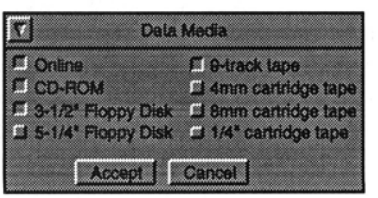


Figure 4: A dialog box to select from an Enumeration

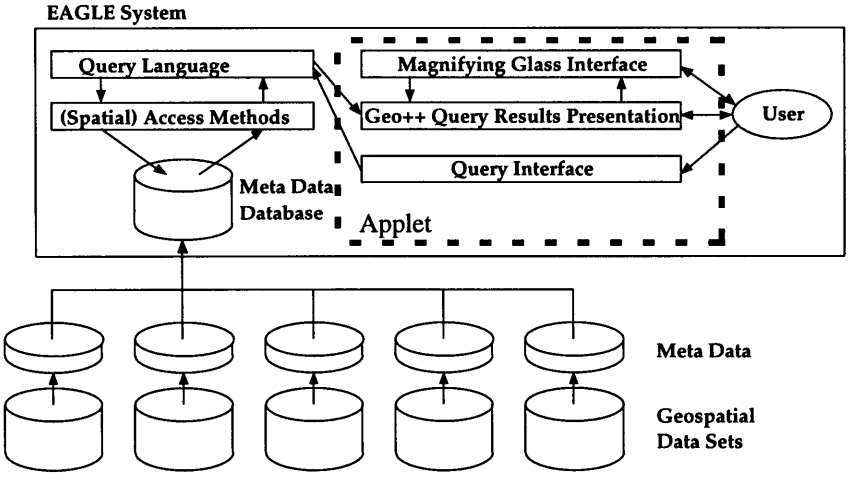


Figure 5: EAGL Architecture

ARCHITECTURE

This section describes an architecture of an environment that woud support the browsing through datasets. Figure 5 shows an overview of the architecture of the EAGL (Exploratory Access to Geographic Libraries) System. The architecture is somewhat similar to the architecture of the Alexandria Project (Smith 1996).

The geographic datasets and their metadata information may be stored on sites all over the world. Since the EAGL system is intended to be a highly interactive tool, it is not possible to use the information stored on these remote sites, even if the required data are on-line. The time to collect the metadata of all the geographic datasets and transfer it to user's site would be too long. Therefore, the metadata has to be collected in advance in a metadata database. Not only is all the metadata available at the fastest speed, but it also allows to install some auxiliary access methods on top of the data. Again, EAGL is designed as an interactive system and needs all the speed it can get. The database must allow for efficient storage of spatial and non-spatial data. Furthermore, it must be able to define indices on both types of attributes. In our implementation we are using Illustra (Illustra 1995) as our metadata database. Illustra is an extendible object relational database on which TNO-FEL has developed a GIS 2D/3D extension similar to the extensions build on top of Postgres (Oosterom 1991).

On top of the database we have developed a front-end application in Java (SUN 96). This applet implements a number of concepts that can be found in GEO++ (Vijlbrief 1993). The applet contains a Main Map View, a Overview Map, buttons for starting the "Layer Manager" and a "Tool Box" in which the magnifying glass can be activated. The applet is being further extended in the

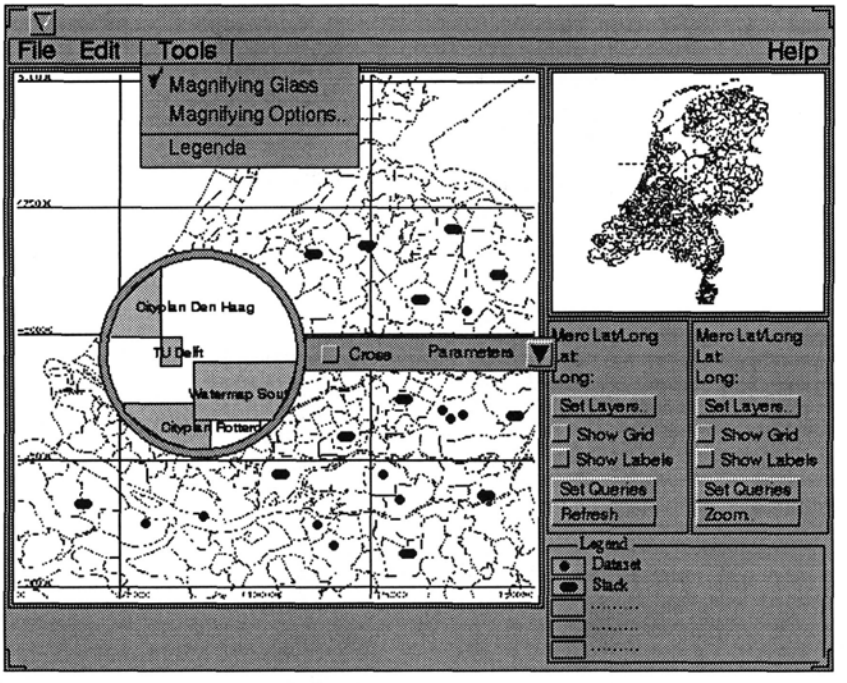


Figure 6: Browsing EAGL with the magnifying glass.

GeoMed project. A project description and design documents can be found at the GeoMed site (GeoMed 1996). Figure 6 shows a browsing session with EAGL.

EXAMPLE SESSION

This section describes a user session with EAGL. The user wants to find a geographic dataset describing a wetlands area in the United States. After starting EAGL, a coarse background world map is shown. Since the user has some constraints on the spatial location of the geographic dataset, zooming in to that area is the first action. Zooming in with EAGL is simply drawing a smaller rectangle on the map. After the right area is on the map window, the user can create a first, probably rough, query. This can be done with the `Set Query` button, which activates a *Query Composer*. The results of this query are presented on top of the background map. Each object in the geographic library represents a dataset covering a certain area. This area is used to display the object on the map. If more than two objects occupy the same area, both objects are replaced by a *stack-symbol*.

The result of the initial query has probably returned a huge amount of objects. The user can now activate the magnifying glass and *browse* through the

result. The user can either decide to exploit the parameters values of the magnifying glass or the user can activate the crosshair and examine the individual datasets.

In this example session, the user decides to use the crosshair and gets information about the digital data format of the datasets. Since the user's Geographic Information System cannot deal with all data formats, the corresponding filter parameter in the magnifying glass is switch on. When the user moves the magnifying glass over the map, with the crosshair switched off, only the objects that qualify the parameter values will be shown. Again, the user gets information about the datasets and is able to set new filter parameters. This step of browsing, interpreting, refining, and again browsing, eventually leads to a very small number of datasets. With the crosshair switched back on, the user can compare the attribute values of each individual object with the magnifying glass parameters and finally decide which datasets to purchase.

CONCLUSIONS

We have presented a tool for exploratory access to geographic libraries. We used the magnifying glass metaphor and combined it with the dynamic filter. So far, there is only a system design and a prototype implementation is under way. A number of open problems will have to answered by the release of this first prototype.

The first problem is what are useful filter parameters. For the design phase we have taken the FGDC metadata standard (FGDC 1997) as a source for filter parameters. In a first selection, we ended up with about 25 possible filter parameters. It is likely that they are not all as useful in the exploring process.

EAGL is designed as a highly interactive system. Some measures have to be taken to reach the required speed. For instance, implementing an advanced search mechanism is necessary (Oosterom 1995).

When a lot of datasets describe the same area, a stack symbol is presented on the map. A problem is how to access each individual element of the stack. One solution might be to move the magnifying glass over the stack and allow the user to flip through it. However, when the stack is very large, this might not be a good solution.

An exploratory access system can become very useful to the user community when public access is allowed. Connecting EAGL to the Internet by means of a World Wide Web (WWW) server seems to be a good idea. A Java implementation would make this possible.

REFERENCES

C. Ahlberg and B. Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, *CHI 94*, 1994, Boston, MA, 313-317.

E. Clementini, A. D'Atri, and P. Di Felice, Browsing in Geographic Databases: An Object-Oriented Approach, *Workshop on Visual Languages*, Skokie, IL, 1990, 125-131.

M. Egenhofer, Why not SQL! *International Journal of Geographical Information Systems* 6(2): 71-85, 1992.

M. Egenhofer and J. Herring, Querying a Geographical Information System, *Human Factors in Geographical Information Systems*, D. Medyckyj and H. Hearnshaw (ed.) Belhaven Press, London, 124-136, 1993.

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

FGDC, Metadata Standards Development, *http://www.fgdc.gov/Metadata/metahome.html*, 1997.

D. Flewelling, Comparing Subsets for Digital Spatial Libraries, *Ph.D. Thesis, Department of Spatial Information Science and Engineering, University of Maine*, 1997.

The GeoMed project, *http://illusion.fel.tno.nl/geomed/geomed.html*, 1996.

P. van Oosterom and V. Schenkelaars, The development of an interactive multi-scale GIS, *International Journal of Geographic Information Systems*, 9(5): 489-507, 1995

P. van Oosterom and T. Vijlbrief, Building a GIS on top of the open DBMS Postgres, *Second European Conference on Geographical Information Sytems*, 775-787, 1991.

A. Sleezer, Direct Manipulation of Temporally Constrained Activities fro Geographic Modeling, *Master's Thesis, Department of Surveying Engineering, university of Maine*, 1994.

T. Smith, A Digital Library for Geographically Referenced Materials, *IEEE Computer* 29(5): 54-60, 1996.

M. Stone, K. Fishkin, and E. Bier, The Movable Filter as a User Interface Tool, *CHI 94*, 1994, Boston, MA, pp. 306-312.

T. Vijlbrief and P. van Oosterom, The GEO++ system: An extensible GIS, *Spatial Data Handling*, Charleston, 1992.

AN INTERACTIVE DISTRIBUTED ARCHITECTURE FOR GEOGRAPHICAL MODELING

Greg A. Wade, Researcher
Department of Computer Science
Southern Illinois University at Carbondale
Carbondale, IL 62901-4511

David Bennett, Assistant Professor and Raja Sengupta, Graduate Student
Department of Geography
Southern Illinois University at Carbondale
Carbondale, IL 62901-4514

ABSTRACT

The creation and modification of geographically explicit models is often difficult and time consuming due to the complexity and scale of the real world systems they simulate. Efficiencies can be gained by sharing models, model components, and data. Recent advances in computing technologies provide new tools that support distributed databases and modelbases. In this paper a distributed, platform independent system for geographic data retrieval and spatial modeling is presented.

1.0 INTRODUCTION

The types of analytical models and tools needed to address spatial problems are often domain specific and not well suited to generic software packages such as geographic information systems (GIS). Furthermore, the creation and modification of geographically explicit models is often difficult and time consuming due to the complexity and scale of the real world systems they simulate. Spatial decision support systems (SDSS) are designed to overcome some of the limitations of current GIS technology (Densham, 1991). However, SDSS that attempt to support the integrated management of geographical data and models are rare. Frameworks for geographical model management have been proposed (e.g., Bennett, in press; Wesseling et al., 1996) but they fail to take advantage of the ever growing capabilities of distributed computing. The work presented here builds on earlier work to create a distributed, platform independent system for geographic data retrieval and spatial modeling. To illustrate the utility of such a system a prototype

distributed Java-based SDSS (DJS) has been developed.

This implementation of a DJS allows decision makers to search network accessible repositories of data using geographical and contextual queries via the Internet or a private intranet. Models and atomic model components may also be retrieved from remote servers. Facilities are provided for linking existing elements retrieved from remote servers with newly created model components to form dynamic systems capable of simulating real-world events. System performance is enhanced by providing for the distribution of the database and modelbase across a network of heterogeneous computers. This allows for the use of divide and conquer techniques to solve computationally intense modeling problems (e.g., some machines can provide traditional GIS facilities while others on the network handle the computational tasks associated with performing complex simulations).

The relatively new computer programming language Java and its extensions contain the functionality needed to implement this project. Java provides a development environment for Internet applications and possesses unique features that facilitate the creation of software designed to be executed in a distributed environment. These features include platform independence and the ability to dynamically load and bind compiled code over the network (Gosling and McGlinton, 1996). Bennett's representational framework for geographical systems was modified and extended to support distributed objects, and its key software elements were ported to Java. Communication protocols and metadata requirements were defined and implemented to support network navigation and cross-platform geographical queries.

2.0 ARCHITECTURE OVERVIEW

The DJS is composed of five components tightly coupled via network communication: object repositories, model repositories, data servers, compute servers, and the DJS console. Figure 1 graphically illustrates the system elements and their relationship to the DJS console. These components may be distributed

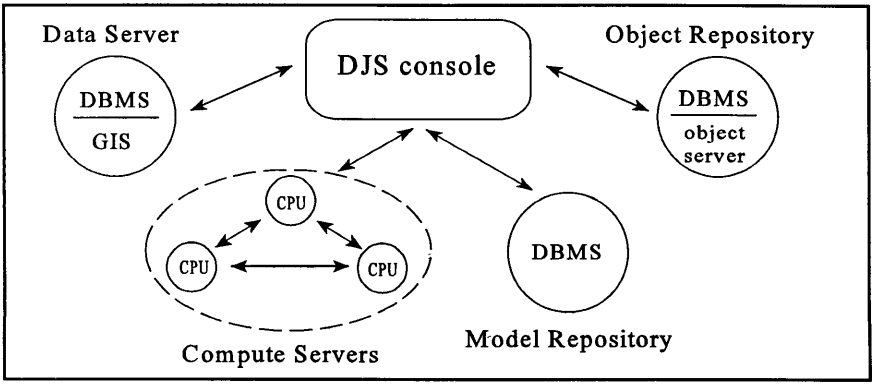


Figure 1. The DJS console's interaction with other system elements.

across heterogenous hosts linked via intranets or the Internet. The DJS console provides the graphical user interface to the system, and is the user's means of interacting with the other four components to create simulation models from model components.

Although each component is discussed as a disjoint entity, the design of the system does not dictate that each system element resides on a separate host. Nor is it required that each system element be on a single host. It is possible that a data server or an object repository may actually consist of several machines acting together as a virtual data server or object repository.

2.1 Object repositories

All data and models within this system are instantiated from Java classes. Class definitions are stored in object repositories that can be accessed across the network. These repositories consist of two elements, a database management system (DBMS) and an object server. The DBMS contains the names of Java classes, the hostname where specific classes are defined, and the TCP port number needed to access these object servers. These servers respond to requests by transferring binary representations of the required Java class over a network connection. This class can then be dynamically loaded into the DJS console or saved to disk for later use.

2.2 Model Repositories

Models are composed of a collection of model components. Each component represents an autonomous element of the simulation, and consists of static inputs, local variables, dynamic inputs, computations and outputs. Static inputs to a component include any value that does not change through the course of the simulation. Local variables and dynamic inputs are allowed to change during the simulation. Local variables are used by model components as state variables and get their values from the components calculations. Dynamic inputs on the other hand get their values from the outputs of other model components. Model repositories are used to locate model specifications or entire models stored in a DBMS. These specifications tell the system what inputs are required by the model or model component and what outputs or results are produced. In addition to locating these components on a model server, the user may create them interactively or load them from the local disk via the DJS console. The list of associated inputs is used to search for "data" needed to run the model. These data can be derived from existing geographic databases or from the output of other model components.

The flow of data from the outputs of one model component to the dynamic inputs of another occurs through *communication channels*. These channels utilize the communication technologies built into compute servers, thus allowing components running on different servers to communicate. The distribution of components can be used to exploit the natural parallelism of some models. Components not dependent upon each other for data can run concurrently as long as they have received all needed dynamic inputs. This form of parallelism follows

from the Dataflow computer architecture.

2.3 Data Servers

Data servers consist of two elements, a DBMS and a GIS. The DBMS is used to initially locate spatial data based on map extent and the data's attributes. In addition to the map extent and attributes for a data set, this database stores metadata required by the DJS. This information includes the hostname of the data server storing the data set, the filename the data is stored in on the server, the type of Java object the data may be loaded into, the individual who created the object definition, the creation date, and modification dates. Figure 2 shows an excerpt from the metadata of a soil's coverage for Jackson County, IL stored in a polygon data structure. After data is located the GIS is used to query and extract data residing on the remote server.

Description:	Soil id's from Jackson County Illinois
Host:	bast.cs.siu.edu
File:	/GeoData/Soil/IL/Jackson
Class:	dms.spatial.poly.PolyLayer
Projection:	UTM
ExtentMinX:	264,000
ExtentMinY:	4,160,000
ExtentMaxX:	311,100
ExtentMaxY:	4,204,100
NumAttributes:	1
Attributes:	Soil-Type

Figure 2. Sample metadata used by the DJS.

If data sets are located that require Java classes not installed locally on the user's computer, an object repository is contacted in an attempt to locate the compiled Java code for the class.

2.4 Compute servers

Compute servers are used to execute models consisting of components built to the user's specifications. Compute servers differ from the other system elements in several ways. They are capable of communicating among themselves while data servers, object repositories, and model repositories are only allowed to communicate with the DJS console. This is needed since model components must often exchange information. Secondly, they do not contain a DBMS. As their name implies compute servers are utilized solely for raw CPU power.

3.0 A SIMULATION EXAMPLE

To illustrate the modeling capabilities of the system, a simple stream network model was developed (Figure 3). This network consists of five segments, s_1 to s_5.

Each segment receives a constant inflow from overland runoff. Additionally segments s_3 and s_5 receive inflow from the outflows of segments s_1, s_2 and s_3, s_4 respectively. From these flows the outflow from segment s_5 is computed over time.

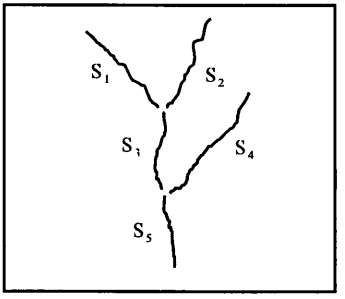


Figure 3. Stream network.

3.1 Simulating stream flow

Equations 1-4 summarize the Muskingum (Linsley et al., 1982) method of channel routing where I_1 and I_2 represent current and previous inflow respectively. Similarly O_1 and O_2 represent current and previous outflow respectively.

$$O_1 = c_0 I_1 + c_1 I_2 + c_2 O_2 \quad (1)$$

$$c_0 = (Kx - 0.5t) / (K - Kx - 0.5t) \quad (2)$$

$$c_1 = (Kx + 0.5t) / (K - Kx + 0.5t) \quad (3)$$

$$c_2 = (K - Kx - 0.5t) / (K - Kx + 0.5t) \quad (4)$$

The coefficients c_0, c_1 and c_2 (equations 2-4) are derived from K the storage constant (i.e., the ratio of storage to discharge), x the relative importance of inflow and outflow in determining storage and t the simulated time interval between computations. For most streams, x is between 0 and 0.3 with a mean value near 0.2 (Linsley et al., 1982). K can be approximated by the travel time through the reach.

3.2 Creating model components

A model component is created to represent each segment. Each component has three static inputs *(K, x,* and *over_land_flow)* which are initialized with values retrieved from the GIS, and an additional static input t supplied by the user. In addition to these inputs each component receives a dynamic input, *flow*, from the output of an upstream component. Any component not having a contribution from upstream flow, i.e., those representing segments s_1, s_2, and s_4, will always receive 0 for *flow*. Components also require the local variables in Table 1.

Variable	Initial Value	Purpose
c_0, c_1, c_2	0	coefficients for the Muskingum method
last_inflow	observed values loaded from the GIS	I_2 for the Muskingum method
last_outflow		O_2 for the Muskingum method
tmp	0	temporary variable

Table 1. Local variables used to simulate stream flow.

The component's variables are used in conjunction with the computations in Figure 4 to compute the components output, *flow*.

$c_0 = (Kx - 0.5t) / (K - Kx - 0.5t)$
$c_1 = (Kx + 0.5t) / (K - Kx + 0.5t)$
$c_2 = (K - Kx - 0.5t) / (K - Kx + 0.5t)$
flow = *flow* + *over_land_flow*
tmp = *flow*
flow = (c_0 * *flow*) + (c_1 * *last_inflow*) + (c_2 * *last_outflow*)
last_inflow = *tmp*
last_outflow = *flow*

Figure 4. Computations used to simulate stream flow.

This simulation could easily be extended by allowing the *over_land_flow* of each segment to vary over time. This value could be determined from the output of another model. The same approach could be taken to create inflows for s_1, s_2 and s_4.

4.0 IMPLEMENTATION DETAILS

As noted earlier the DJS was coded in Java. Development was done using Sun Microsystem's Java Development Kit (JDK) and Symantec's Café. This implementation utilizes four different programming techniques and tools.

4.1 Accessing the DBMS

The Java Database Connectivity (JDBC) API was used to access DBMS throughout the system. JDBC provides a convenient interface between the Java programming language and SQL databases. The JDBC API is implemented as a set of driver managers that can connect to various databases (Sun Microsystems, Inc., 1996). The JDBC driver chosen for the DJS implementation is a commercial product marketed by XDB Systems. XDB Systems' JetConnect JDBC driver in conjunction with their jetport server provide a bridge between JDBC and databases compliant with Microsoft's Open Database Connectivity (ODBC) standard (Ball et al., 1996).

Client software (i.e., the DJS console) using JDBC and XDB Systems' JDBC driver connect to the jetport server on a remote host, i.e., a machine acting as a data server, object repository, or model repository. The jetport daemon then accesses a Microsoft Access database via ODBC to perform database functions. See Figure 5.

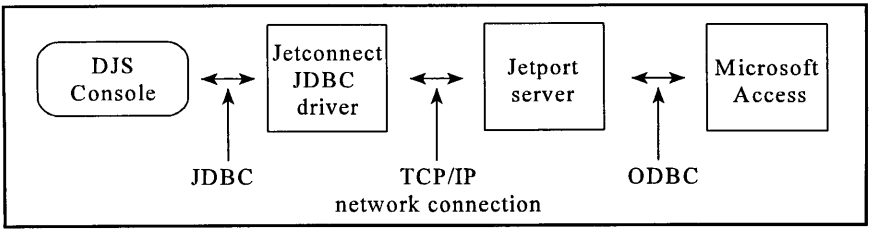


Figure 5. The DJS console's interaction with the data server's DBMS.

4.2 The object server

New models located by the search engine may require Java objects that are unknown to the computation host. As such, a mechanism must exist to export class definitions to compute servers and the DJS console. This is the role of the object server. To construct the object server a custom *ClassLoader* was created that is used to dynamically load compiled class files into the Java virtual machine. When Java classes, or objects instantiated from them, attempt to access an unknown class the class loader is invoked to locate the needed object code (Gosling et al., 1996). This allows the DJS console to load a class when a user attempts to access a data set requiring a spatial structure not available locally. The DJS console also implements a cache as part of its class loader to reduce the overhead of repetitively searching for and downloading classes.

4.3 Accessing the remote GIS

There does not currently exist an API similar to JDBC and ODBC for accessing remote GIS software. As a result, a radically different approach was used for accessing the GIS component of the data servers than was used for accessing the DBMS. However, the results are similar. In both cases the client sends a request, either an SQL or GIS query, and a result is returned. The GIS software is not accessed via an API. Rather, it is accessed by instantiating Java objects representing spatial data structures on the data server. A data file on the server is then loaded into the object. These remote objects are provided by HORB. HORB extends Java to provide object request broker capabilities that allow clients to create remote instances of objects and execute their methods on a remote machine across the network (Hirano, 1996). As such, the data is not transferred to the host running the DJS console. HORB accomplishes this through the use of proxy objects on the client. These proxy objects use the HORB library to communicate with a HORB server (Figure 6).

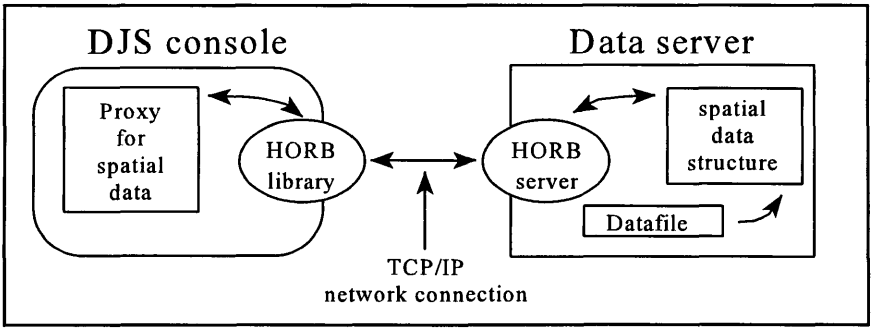


Figure 6. The DJS console's interaction with the GIS of a data server.

4.4 Creating model components and communication channels

The system also utilizes HORB to create model components on the compute servers. A model component object is instantiated on the compute server and methods are called to initialize its static inputs, local variables, dynamic inputs, computations and outputs. The component can then be started and stopped remotely. In addition to the model component, HORB is used to create a communication channel for each model component having outputs. This channel is created on the same compute server as the component sending outputs to the channel. Other components, again using HORB, then access this channel to retrieve dynamic inputs.

5.0 CURRENT LIMITATIONS AND FUTURE WORK

As with any prototype software system, this implementation of a DJS has several limitations. Most of these limitations all have solutions based in current geographic and computing technologies. Enhancing the current system through these technologies paves the way for future research.

5.1 The GIS

First the spatial data handling component of the data servers is primitive. Currently two types of spatial data structures, a vector and raster structure have been implemented. While these structures allowed for testing of the system, they provide only simple data retrieval. This limitation might be corrected by incorporating the work of Sengupta et al. (1996) which focuses on developing intelligent agents capable of interfacing with existing GIS products. This would allow the system to leverage the investment in pre-existing software and datasets. In implementing this approach, API's similar to JDBC and ODBC could be created for accessing GIS via these agents from Java.

5.2 Computational ability of model components

Currently the computational ability of model components is limited. Model components are only equipped to handle calculations consisting of basic

mathematical operations. However most real-world simulations require more advanced calculations. To resolve this problem a commercial package such as Mathematica or Maple could be used. The solution has one disadvantage: platform independence would be lost. An ideal solution would be to incorporate a Java-based symbolic mathematics library. Unfortunately, as of this writing, the authors know of no such package.

5.3 Increasing system performance

Lastly system performance needs to be improved. More of an effort needs to be done to exploit the parallelism of models. At present the DJS console simply cycles through its list of compute servers when creating model components, assigning a component to the next server in the list. The Vertically Layed (VL) allocation scheme to determine which model components to assign to which compute servers should be implemented. The VL algorithm uses heuristic rules in an attempt to maximize parallelism while minimizing communication overhead (Hurson et al., 1990). This would have the result of assigning model components capable of executing concurrently on different compute servers while assigning components which must execute sequentially to the same compute server. To even further enhance performance an optimization phase for the VL allocation scheme purposed by Kvas et al. (1994) could be added. This phase would take into account communication delays between compute servers. This would be essential for achieving high performance when utilizing compute servers spread across the Internet.

6.0 CONCLUSION

This implementation of a prototype DJS has illustrated the feasibility and functionality of a distributed platform independent geoprocessing SDSS. Current software technologies and techniques have matured to the point where creating such a system is possible. As a result a DJS can overcome some of the problems associated with current SDSS. In addition it can bring the power and flexibility of SDSS to many users who do not currently possess high-end computing resources. These users can run the DJS console on almost any personal computer or workstation supporting the Java virtual machine, and then access high performance computing facilities provided in house via an intranet or globally via the Internet.

ACKNOWLEDGMENTS

This work was funded in part by a grant from the Pontikes Center for the Management of Information, Southern Illinois University at Carbondale, Carbondale, IL 62901.

REFERENCES

Bennett, D.A. (in press). A Framework for the Integration of Geographic

Information Systems and Modelbase Management. *International Journal of Geographical Information Systems.*

Densham, P.J. (1991). Spatial decision support systems, in *Geographical Information Systems: Principles and Application*, ed. Maguire, D.J., Goodchild, M.F., and Rhind, D.W., Longman, London, pp. 403-412.

Gosling, J. and McGlinton H. (1996). *The Java Language Environment: A White Paper.* Sun Microsystems Inc.

Gosling, J., Yellin, F., and The Java Team. (1996). *The Java Application Programming Interface, Volume 1.* Addison-Wesley, pp. 19-22.

Hirano, S. (1996). What's HORB?. *http://ring.etl.go.jp/openlab/horb/doc/what.htm.*

Hurson, A.R., Lee, B., Shirazi, R., and Wang, M. (1990). A Program Allocation Scheme for Data Flow Computers. *Proceedings of the 1990 International Conference on Parallel Processing*, pp. I-415-I-423.

Ball, K., McClain, D., and Minium, D. (1996). *Bringing a New Dimension to Java Through Easy Access to Enterprise Data.* XDB Systems, Inc.

Kvas, A., Ojsteršek, M., and Žumer, V. (1994). Evaluation of Static Program Allocation Schemes for Macro Data-Flow Computer. *Proceedings of the 20th EUROMICRO Conference*, IEEE Computer Society Press, pp. 573-580.

Linsley, Jr., R.K., Kohler, M.A., and Paulhus, J.L.H. (1982). *Hydrology for Engineers.* McGraw-Hill, pp. 275-277.

Sengupta, R. R., Bennett, D.A., and Wade, G.A. (1996). Agent Mediated Links Between GIS and Spatial Modeling Software Using a Model Definition Language. *GIS/LIS '96 Annual Conference and Exposition Proceedings*, pp. 295-309.

Sun Microsystems, Inc. (1996). *JDBC Version 1.1 Release Notes.*

Wesseling, C.G, Karssenberg, D., Burrough, P.A., van Deursen, W.P., 1996, Integrating dynamic environmental models in GIS: The development of a dynamic modelling language. *Transactions in GIS*, 1(1):40-48.

No Fuzzy Creep!
A Clustering Algorithm for Controlling Arbitrary Node Movement

Francis Harvey
EPFL-IGEO-SIRS
GR-Ecublens
CH-1015 Lausanne
Switzerland
Francis.Harvey@dgr.epfl.ch

François Vauglin
IGN-COGIT Laboratory
2 avenue Pasteur
F-94160 Saint-Mandé
France
Francois.Vauglin@ign.fr

ABSTRACT

A perennial problem in vector overlay is fuzzy creep. Commercial vector overlay algorithms resolve near intersections of lines employing arbitrary node movement to align two chains at nodes selected randomly in the area of an epsilon band. While this solution is effective in reducing the number of sliver polygons, it introduces distortion. In some situations this distortion may be tolerable, but in others it may produce positional errors that are unacceptable for the cartographic or analytical purpose. Our research aims to provide an extension of overlay processing that provides a solution for GIS uses that require more exact control over node movement. The key to this is a robust, non-distorting cluster analysis. The cluster algorithm we present fulfills two goals: 1) it selects nodes based on an nearness heuristic, 2) it allows the user to fix the position of one data set's nodes and moves the other data set's nodes to match these position. In this paper we review existing cluster algorithms from the computational geometry and analytical cartography literature, evaluating their heuristics in terms of the potential to avoid fuzzy creep. Grouping the algorithms into a bit-map and fuzzy-detection types, we discuss the advantages and disadvantages of each approach for controlled near intersection detection. Based on the results of this analysis, we present a algorithm for non-distortive geometric match processing, the basis for our work on geometric match processing.

1. THE PROBLEM WITH FUZZY CREEP

Vector overlay is utilized for a diverse range of purposes to combine geographic information. These purposes place numerous positional accuracy demands that we find are only partially served by existing vector overlay algorithms. All geographic data contains some positional inaccuracy, processing should not increase inaccuracy. We find there is ample need for vector processing algorithms that provide more control.

A crucial problem in current vector overlay algorithms is fuzzy creep (Pullar, 1990; Pullar, 1991; Pullar, 1993; Zhang & Tulip, 1990). Fuzzy creep is

the arbitrary movement of nodes during overlay processing resulting from node snapping, centroid assignment, and induced intersections (Pullar, 1991). This is the result of using a fuzzy tolerance (also known as epsilon tolerance) to resolve near intersections, that otherwise can turn into splinter (or spurious) polygons. Because of its great advantages for resolving near intersections and numerical inaccuracies in processing this type of vector overlay, more commonly known as fuzzy vector overlay (Guevara & Bishop, 1985), is the most common algorithm. Without this algorithm, overlay would be encumbered by a vast amount of spurious polygons, greatly inhibiting the analytical potential of this quintessential GIS operation (Goodchild, 1978).

Still, in spite of great utility, vector overlay algorithms may introduce undesired side effects. These issues are especially pertinent for purposes that require a more exacting control of the overlay operation, especially in terms of positional accuracy. Current applications of fuzzy vector overlay can introduce arbitrary movement of geometric features up to, or even greater, than the epsilon tolerance (Pullar, 1993; White, 1978). The limitation to one fuzzy tolerance for all data sets reduces control possibilities yet further.

We have addressed this broad set of problems in earlier work (1994, 1996) on geometric matching, and in this paper present the continuation of our efforts with a focus on cluster analysis. Briefly, this work has already outlined an algorithm for controlling the movement of nodes by employing multiple tolerances. We distinguish between a match tolerance for the more accurate data set, and a shift tolerance for the less accurate data set. It is possible to align features (without any loss of positional accuracy) from the less accurate data set with features from the more accurate data set.

As the number of digital data sets grows we expect to find an increase of the situation when accurate digital geographic data is combined with less accurate data, i.e. situations when data from field notebooks is combined with digital topographic data, or remote sensing data is combined with precision survey data. A multitude of applications will require functions that combine data, but retain the positional accuracy of the most accurate data set.

Cluster analysis "organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups" (Jain & Dubes, 1988). It is crucial to controlling fuzzy creep in vector overlay processing, and has already received attention in analytical cartography. Zhang and Tulip (1990) point to the potential loss in positional accuracy due to fuzzy creep. Pullar (1993), at AutoCarto 11, describes how naively implemented clustering algorithms can lead not only to considerable loss of positional accuracy through fuzzy creep, but even the actual disappearance of features. Beyond his proposal for a technique to control cumulative arbitrary node movement, we present in this paper an algorithm for limiting, and in certain cases, eliminating fuzzy creep.

In the next section we describe existing clustering algorithms used in vector overlay. The following section describes the requirements geometric matching places on clustering algorithms. The conclusion and summary discuss the results, further steps in our research, and possibilities for additional developments.

2. EXISTING CLUSTERING ALGORITHMS

In this section we review existing clustering algorithms for vector overlay. We focus on on each algorithm's maintenance of positional accuracy, computational demands, and suitability for implementation in fuzzy overlay processing.

Approaches using integer and rational arithmetics (Cook, 1978; Franklin, 1987; Wu & Franklin, 1990) have their merits, but computational demands restrict their usefulness for our purposes. Earlier, computer resources were limited and the fuzzy vector overlay using a band-sweep approach became the most successful and common technique. The band-sweep approach means that only a subset of each data set is loaded in memory at a time (White 1978) improving processing efficiency enormously. The reduction in memory demands, the ability to profit from the speed of floating point calculations, and deal with computational imprecision led to its success.

We distinguish between two basic approaches to clustering in computational geometry, bit-map and vector. Here we focus our examination of clustering algorithms on avoidance of fuzzy creep, general computational complexity, and ease of implementation in fuzzy vector overlay processing. First, we will review bit-map approaches.

We identify three bit-map approaches: continuous relaxation, coincidence calculation, and rasterized vector grid. Continuous relaxation is a technique that comes from remote sensing to find correspondences to vector data bases. It aims to construct homogeneous segmented areas based on, a priori probabilities, a proximity relation, a compatibility function, and the definition of an influence function (Lemarié & Raynal, 1996). It is effective for detecting changes in a vector data base, but is very complex and very sensitive to the chosen parameters. It is also very computationally complex and not easily optimized. This approach was developed for raster/vector comparisons, its utility for the cluster analysis of vector data sets remains questionable without further work.

Coincidence calculation actually consists of different techniques. The similarity in these techniques is the basic calculation method. After overlapping polygons (or raster areas) are identified, the similarity parameter is determined using the areas. The similarity parameter gives a probability that the two polygons are the same. A distance parameter can also be calculated. This gives

the relative distance between the areas. The major problem with this approach for our purposes is its limitation to areas.

The rasterized vector grid approach begins with vector data. This data is rasterized into a defined grid and then either the continuous relaxation or coincidence calculation approach is used for cluster analysis.

The restraint to areas in the bit-map approaches is a considerable problem. If the resolution of the raster corresponds to the known accuracy of the data set, these approaches may be quite valuable. However, in cases when the positional accuracy is greater than the cell resolution, rasterizing artificially limits accuracy to the cell size. Fuzzy creep would remain an issue in these cases. In any case, because fuzzy vector overlay processing is not designed around the algorithmic requirements of bit-map cluster analysis, considerable computational inefficiencies could result from implementing these approaches.

Vector approaches to cluster analysis generally allow more exact control over the analysis, but are computationally more complex. Their largest advantage for vector overlay is the ease of integration into existing vector overlay processing algorithms. Vector cluster analysis rests on proximity analysis merging clusters until a condition is met (Jain & Dubes, 1988).

Milenkovic (1989) presents the simplest approach to clustering vector data. His method merely tests if any points are found within an epsilon band, if so they are merged. This operation is repeated until there are no more nodes within the epsilon band. Of course, this leads to a high potential for fuzzy creep.

The approach to clustering in Odyssey's Whirlpool overlay processor, does a somewhat better job. Although the problem of fuzzy creep is recognized, it still allows arbitrary node movement (Chrisman, Dougenik, & White, 1992; White, 1978). Based on this work and others, other approaches were proposed that strive to control fuzzy creep. Zhang and Tulip (1990) specifically address the problem of induced intersections that result from arbitrary node movement. Their approach is based on a proximity matrix that relates objects and the analysis of the matrix uses hierarchical classification. All nodes in the same epsilon band are candidates for merging. Only nodes whose epsilon bands overlap reciprocally are merged. This effectively controls fuzzy creep to the extent of the overlapping epsilon tolerances.

David Pullar proposed an approach similar to Zhang and Tulips (Pullar, 1993). Most notably he describes several constraints that define the clustering. First, a new point must be within the epsilon tolerance of an existing node. Second, to be merged, a node must lie within the epsilon tolerance of the cluster center. Third, in the resulting data set two points cannot share an epsilon tolerance. The constraints are very valuable, but maintaining them in his described implementation is very difficult

These constraints and approaches provide the most substantial base for our development of a clustering algorithm that controls fuzzy creep. Because of its affinity to cartographic data processing, our implementation will build on Pullar's constrained clustering algorithm. In the next section we will look at our requirements for cluster analysis in geometric match processing in detail.

3. GEOMETRIC MATCHING REQUIREMENTS FOR CLUSTERING ALGORITHMS

The requirements for a clustering algorithm that supports geometric matching are few and simple:

1) It eliminates or minimizes fuzzy creep.
2) It gives precedence to the more accurate (match) data set.
3) It selects nodes to merge based on a nearness criteria.
4) The match tolerance may not be greater than the smallest distance between nodes in the match data set.
5) It merges all nodes in each cluster.

In the examples we have thought of, we frequently end with cases that can only be resolved by considering semantic information that at the moment is not available in geometric match processing. This is the most serious caveat for this method and need further work. As geometric match processing stands, it cannot successfully resolve all intersections. The basic limitation is the distance between match and shift nodes. If the shift tolerance is greater than the distance between nodes, they will merge during processing.

Only the judicious application of match and shift tolerances can fulfill the requirements with the clustering algorithm we propose here. To eliminate fuzzy creep, the match tolerance must be set to zero, if it is greater, fuzzy creep is only limited to the value of the match tolerance. Furthermore, the more accurate data set must receive the match tolerance. Following the nearness criteria leads to a trade-off between false positives and complete merging.

The match tolerance must be smaller than the smallest distance between match data set nodes. This is necessary to prevent the creep or collapse of whole clusters. It is also necessary that all nodes in each cluster be merged to assure complete resolution of all possible node merges.

4. CLUSTERING FOR GEOMETRIC MATCHING

Following the band-sweep approach to overlay, we now present a clustering algorithm for use in geometric matching.

Vector overlay using the band-sweep algorithm consists of five steps (White, 1978). Cluster analysis is necessary in several of these steps. Basically, after breaking the chains down into monotonic segments cluster analysis is called for to merge nodes.

Based on Pullar's constrained clustering algorithm (1993), our clustering algorithm distinguishes itself by the priority by which it evaluates match data set nodes. This enhances Pullar's algorithm, and we believe this makes it a must more useful technique. First, if a overlay has to consider $n_m + n_s$ nodes as clustering center points, clustering for geometric matching needs only process match data set nodes. Shift data set nodes are always cluster elements, never cluster centers. Further, because of the requirement that match tolerance be greater than the smallest distance between match nodes, no selection of cluster centers is ever required.

There are several rules (constraints) we have set down to describe clustering behavior and avoid erroneous results.

1) No node can be moved a distance greater than its match or shift tolerance.
2) Nodes with overlapping match tolerances cannot be merged. This would violate the second requirement. If this condition is encountered, processing will terminate and pertinent information provided for the user.
3) When the shift tolerances of two nodes overlap, they may only be merged if the Euclidean distance between them is less than the shift tolerance.
4) Merging of nodes is based on proximity to the cluster center.

The cluster algorithm also considers induced and exact intersections resulting from cluster processing. The band-sweep approach has great benefits for dealing with the large number of intersections that can be created during processing.

Different tolerances leads to different potential clustering situations. Figures 1 and 2 present input and output for four different situations grouped by tolerance value ranges. Figure 1 shows the input situations and figure 2 the output of different clusterings. The two left-side groups depict situations with the shift tolerance set to zero. The upper groups illustrate situations when the match tolerance is set to zero. In the lower groups, the match tolerance is greater than zero, on the right the shift tolerance is greater than zero. In all of the eight situations on the left side of the figures, nothing changes during processing. When the tolerances allow movement, changes occur. The normal case for geometric match processing, when the match data set is more accurate than the shift data set, is illustrated in the lower right group. The specific case, when the geometric matching is used to align elements is illustrated in the upper right group.

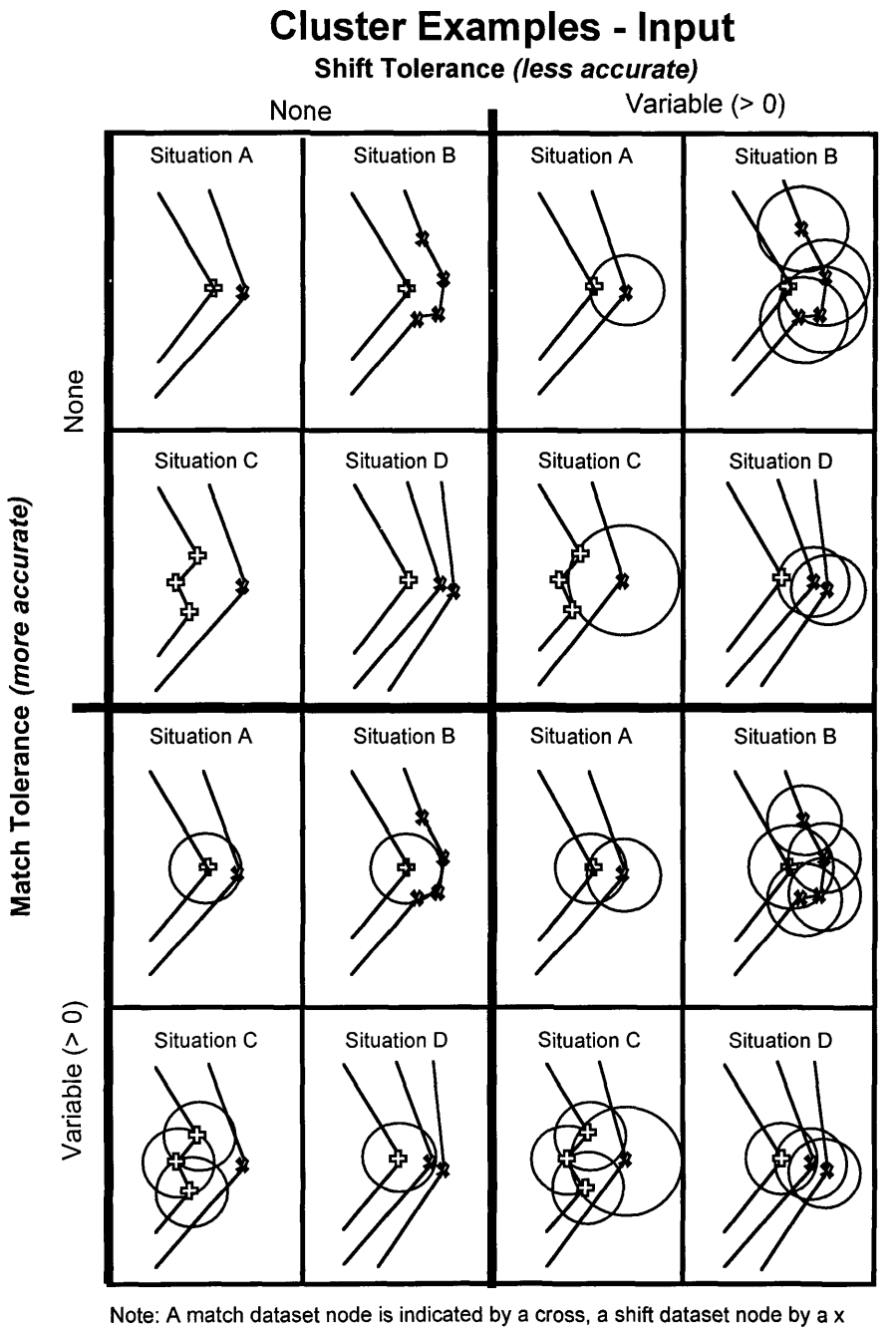


Figure 1 Examples for clusters applying multiple tolerances (Input)

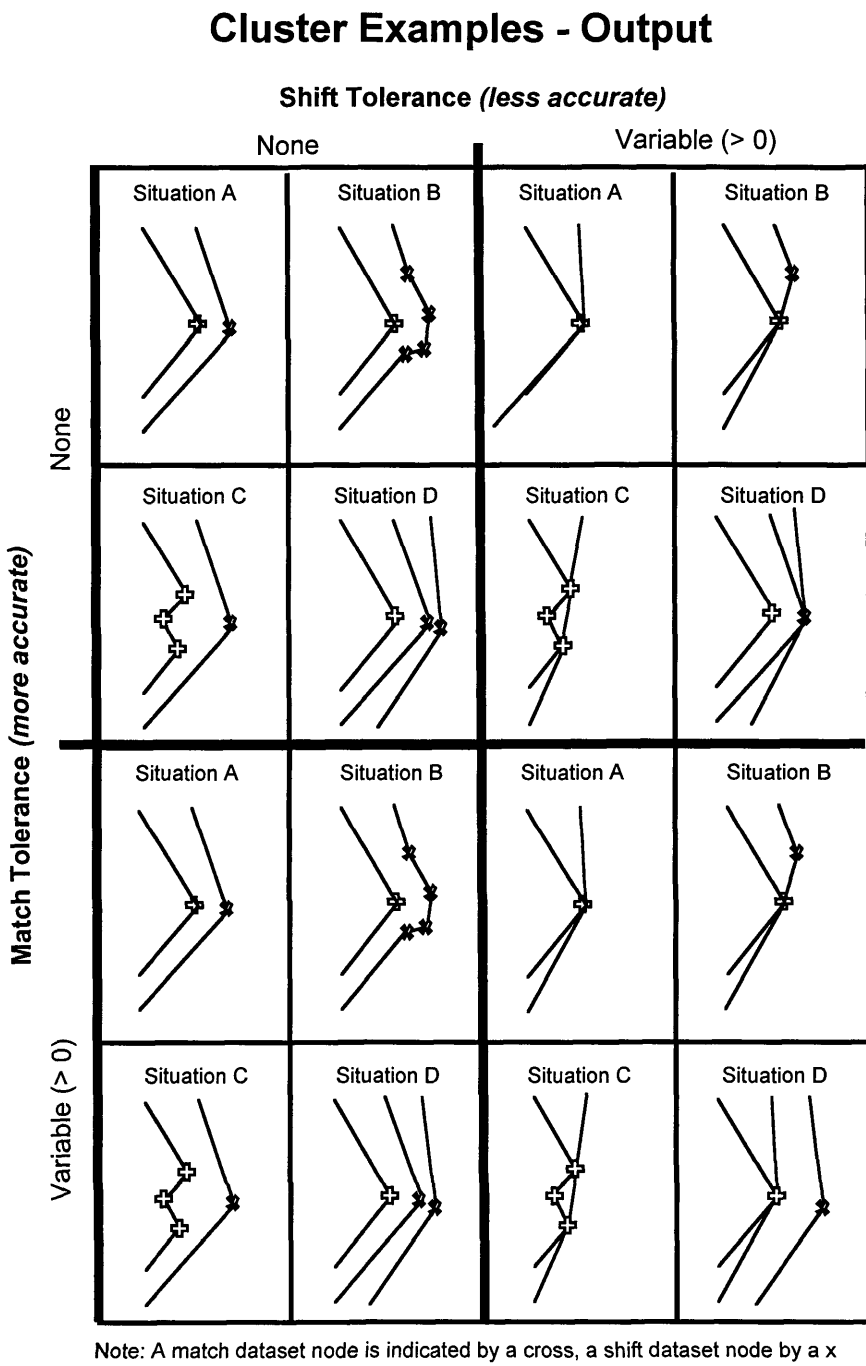


Figure 2 Examples for clusters applying multiple tolerances (Output)

5. SUMMARY AND CONCLUSION

Consideration of fuzzy creep and our implementation of Pullar's algorithm shows that we have only been partially successful in our goal of eliminating fuzzy creep. In summary, Pullar's constrained clustering algorithm reduces possible fuzzy creep to the epsilon value. In geometric matching this is revised to the value of the match tolerance. If the match tolerance is zero, the more accurate data set nodes remain at their locations no fuzzy creep is introduced. As the match tolerance value increases, the amount of fuzzy creep does too. It is impossible to eliminate fuzzy creep if node movement is allowed. In any case, rounding effects and precision limitations will always affect geometric match processing on floating point computers. This issue is only pertinent for extremely accurate work and needs due consideration where the utmost in accuracy is desired.

We conclude the best type of cluster analysis to limit arbitrary node movementfor geometric matching follows vector clustering algorithms. There are several strong reasons for this conclusion:

- it fits within fuzzy overlay processing
- clustering is integral to matching
- extension of existing algorithms
- it can be extended to include feature based parameters

Clearly this approach to clustering in the context of geometric match processing is still limited. We find there is still need to consider statistical methods for determining accuracies and resolving clusters, include feature-based semantics in the geometrification of cluster analysis, preserves shapes, and preserve directions.

At this point we are only able to complete a broad-brush cluster analysis. At least we preserve topology. Based on this work, we believe further improvements will require feature-orientated cluster analysis. Our first thoughts in this direction lead us to consider adding information to the vector data set, for example extending the data structure to include the original x and y locations and the feature epsilon tolerance (x_c, y_c, ε).

References

Chrisman, N. R., Dougenik, J., & White, D. (1992). Lessons for the design of polygon overlay processing from the Odyssey Whirlpool algorithm. In International Symposium on Spatial Data Handling, Proceedings, . Charleston, NC: SDH.

Cook, B. G. (1978). The Structural and Algorithmic Basis of a Geographic Data Base. In G. Dutton (Eds.), Harvard Papers on GIS, First International

Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Cambridge: Harvard University.

Franklin, W. R. (1987). A polygon overlay system in prolog. In AutoCarto 8, Proceedings, Vol. 1 (pp. 97-106). Baltimore, MD: ACSM.

Goodchild, M. F. (1978). Statistical Aspects of the Polygon Overlay Problem. In Harvard Papers on GIS, First International Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Cambridge: Harvard University.

Guevara, J. A., & Bishop, D. (1985). A fuzzy and heuristic approach to segment intersection detection and reporting. In AutoCarto 7, Proceedings, 1 (pp. 228). Washington D.C.

Harvey, F. (1994). Defining unmoveable nodes/segments as part of vector overlay. In T. C. Waugh & R. G. Healey (Ed.), Sixth International Symposium on Spatial Data Handling, 1 (pp. 159-176). Edinburgh, Scottland: T. C. Waugh IGU Commision on GIS/Association for Geographic Information.

Harvey, F., & Vauglin, F. (1996). Geometric match processing: Applying Multiple Tolerances. In M. J. Krakk & M. Molenaar (Ed.), The Seventh International Symposium on Spatial Data Handling (SDH'96), Proceedings, Vol. 1 (pp. 4A-13 - 4A-29). Delft, Holland: International Geographical Union (IGU).

Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall.

Lemarié, C., & Raynal, L. (1996). Geographic data matching: First investigations for a generic tool. In GIS/LIS '96, Proceedings, 1 (pp. 405-420). Denver, Co: ASPRS/AAG/URISA/AM-FM.

Milenkovic, V. J. (1989). Verifiable implementations of geometric algorithms using finite precision arithmetic. In D. Kapur & J. L. Mundy (Eds.), Geometric Reasoning (pp. 377-401). Cambridge, MA: MIT Press.

Pullar, D. (1990). Comparative study of algorithms for reporting geometrical intersections. In K. Brassel & H. Kishimoto (Ed.), Fourth International Symposium on Spatial Data Handling (SDH), Proceedings, Vol. 1 (pp. 66-76). Zürich: Waugh, T. IGU/AGI.

Pullar, D. (1991). Spatial overlay with inexact numerical data. In AutoCarto 10, Proceedings, 1 (pp. 313-329). Baltimore, MD: ACSM.

Pullar, D. (1993). Consequences of using a tolerance paradigm in spatial overlay. In R. McMaster (Ed.), AutoCarto 11, Proceedings, 1 (pp. 288-296). Minneapolis, Minnesota.

White, D. (1978). A Design for Polygon Overlay. In Harvard Papers on GIS, First International Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Harvard University.

Wu, P. Y. F., & Franklin, R. W. (1990). A logic programming approach to cartographic map overlay. Computational Intelligence, 6(2, May 1990), 61-70.

Zhang, G., & Tulip, J. (1990). An algorithm for the avoidance of sliver polygons and clusters of points in spatial overlay. In Fourth International Conference on Spatial Data Handling (SDH), Proceedings, (pp. 141-150). Zurich:

Maintaining Consistent Topology including Historical Data in a Large Spatial Database

Peter van Oosterom
Cadastre Netherlands, Company Staff
P.O. Box 9046, 7300 GH Apeldoorn, The Netherlands.
Phone: +31 55 528 5163, Fax: +31 55 355 7931.
Email: oosterom@kadaster.nl

This paper describes a data model and the associated processes designed to maintain a consistent database with respect to both topological references and changes over time. The novel contributions of this paper are: 1. use of object identifiers composed of two parts: oid and time; 2. long transactions based on a check-out/check-in mechanism; and 3. standard SQL (structured query language) enhanced with SOL (spatial object library) for both the batch production of update files and for the interactive visualization of the changes over time.

1 Introduction

Large scale Topographic and Cadastral data in the Netherlands [9] are stored and maintained in *one integrated system* based on the relational database CA-OpenIngres with the spatial object library (SOL) [4] and X-Fingis [10, 11, 13]. Storing and maintaining consistent topological relationships is important in a spatial database. Topology is essential to the nature of the Cadastre: parcels may not overlap and parcels should cover the whole territory. About 400 persons (surveyors, cartographers) are updating these data simultaneously. After the initial delivery of all data, the customers get periodic updates of the database. Without storing object-history in the database, these *update files* are difficult to extract [16]. Historical data is also used to find the previous owners of a certain polluted spot. This illustrates the need for consistently maintaining both time and topology in the database.

General introductions to spatio-temporal modeling are given in [14, 18, 21][1]. Although several authors have described a spatial-temporal data model and query language, they ignore the problem of maintaining the data in their models, which is complicated due to the topology references. Our data model based on topology and history is presented in Section 2. Topological editing of information is discussed in Section 3, in which particular attention is paid to the fact that multiple users must be able to work simulta-

[1] A glossary of temporal terms in databases can be found in [8].

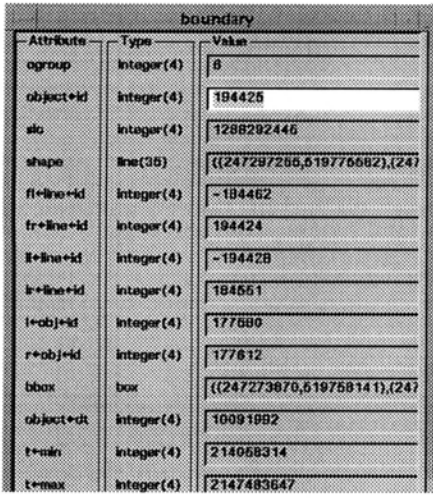


Fig. 1: Boundary record 194425

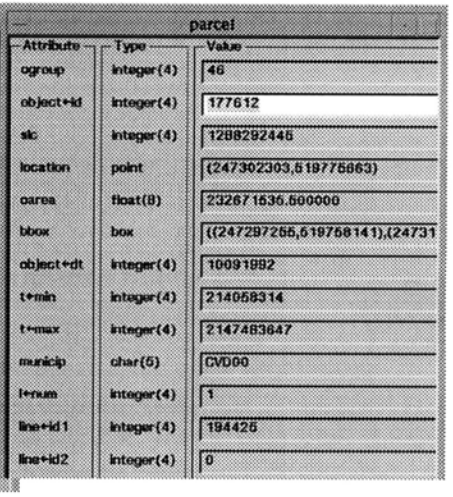


Fig. 2: Parcel record 177612

neously. The production of update files using standard SQL (structured query language) is described in Section 4. In contrast to these 'batch' type of jobs, some possibilities for interactive visualizations of changes over time are given in Section 5 together with other future work. Finally, conclusions can be found in Section 6.

2 Data model

Integrated storage of all components of the data (metric information, topology, thematic attributes, and historic information) in one database is the key property, which enables controlling data consistency. Example records are shown in Fig.1 and 2: `boundary` with parcel boundaries and `parcel` with additional parcel information. Note the integrated use of traditional data types and spatial data types, such as `point`, `line`, and `box` in the data model. In the data model all objects get a unique identifier `object_id`[2], which enable efficient communication with customers of the update files.

Topological references

In theory, explicitly storing planar topological information (references) causes data redundancy, because the references can be derived from accurate metric information as stored in the `shape` attribute of type `line(35)` in the `boundary` table and in the `location` attribute of type `point` in the `parcel` table. However, explicitly storing the topological references makes checking the topological structure (data quality) feasible within the database. Further, it is also convenient for data manipulation; e.g. compute the polygon [3] or find neighbors of a face.

[2]The `object_id` is unique within each group of an object type `ogroup` and is maintained nation-wide. Sometimes in this paper the pair `ogroup, object_id` is abbreviated to just `oid` for simplicity.

[3]The terms *face*, *edge*, and *node* are

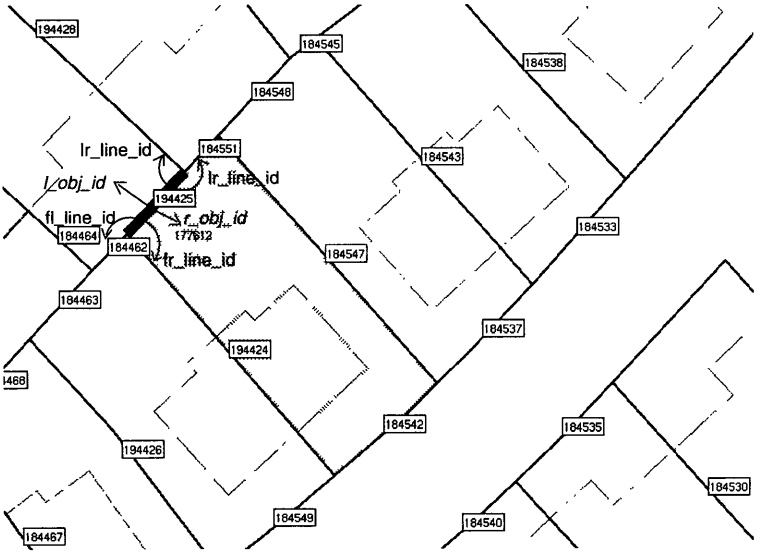


Fig. 3: GEO++ screendump with example boundary record

The spatial basis of the data model is a planar topological structure, called the *CHAIN-method* [15], similar to the *winged edge structure* [3]; see Figs. 1, 2, and 3. However, all references to edges are *signed* (+/−), indicating the direction of traversal when forming complete boundary chains. The edges contain four references to other edges: in the `boundary` table there are attributes to indicate the immediate left and right edge at the first point (`fl_line_id` and `fr_line_id`) and the immediate left and right edge at the last point (`ll_line_id` and `lr_line_id`). Further, references from a face to the first edge of its boundary chain and, if islands are present, references to the first edge of every island-chain are stored. In this model polygons related to faces can be composed by using the signed references only. So, without using geometric computations on the coordinates. Besides the references from faces to edges, and from edges to edges, there are also references from edges to left and right faces: `l_obj_id` and `r_obj_id` in the `boundary` table. A bounding box `bbox` attribute is added to every table with spatial data in order to implement efficient spatial selection. Finally, the computed area is stored in the `oarea` attribute of the `parcel` table.

Historical information

The updates in our database are related to changes of a discrete type in contrast to more continuous changes such as natural phenomena or stock rates. The number of changes per year related to

used when the topological aspects are intended. The terms *polygon*, *polyline*, and *point* are used when discussing the metric aspects. Finally, terms such as *parcel* and *boundary* are used to refer to the objects.

the total number of objects is about 10%. It was therefore decided to implement history on tuple level[4]. This in contrast to implementing history on attribute level, which requires specific database support or will complicate the data model significantly in a standard relational database; see [19, 14, 20, 27]. In our model every object is extended with two additional attributes: `tmin` and `tmax`[5]. The object description is valid starting from and including `tmin` and remains valid until and excluding `tmax`. Current object descriptions get a special value `MAX_TIME`, indicating that they are valid now. `MAX_TIME` is larger than any other time value. There is a difference between the *system (transaction)* time, when recorded object changed in the database, and the *valid (user)* time, when the observed object changed in reality. In the data model `tmin/tmax` are system times. Further, the model includes the user time attribute `object_dt` (or `valid_tmin`) when the object was observed. Perhaps in the future also the attributes `last_verification_dt` and `valid_tmax` could be included, which would make it a *bitemporal* model.

When a new object is inserted, the current time is set as value for `tmin`, and `tmax` gets a special value: `MAX_TIME`. When an attribute of an existing object changes, this attribute is not updated, but the complete record, including the `oid`, is copied with the new attribute value. Current time is set as `tmax` in the old record and as `tmin` in the new record. This is necessary to be able to reconstruct the correct situation at any given point in history. The *unique identifier* (key) is the pair `(oid, tmax)` for every object version in space and time.

For the topological references, only the `oid` is used to refer to another object and not `tmax`. In the situation that a referred object is updated and keeps its `oid`, then the reference (and therefore the current object) does not change. This avoids, in a topologically structured data set, the propagation of one changed object to all other objects as all objects are somehow connected to each other. In case the `oid` of a referred object has changed (becomes a different object), the referring object is also updated and a new version of the referring object is created.

The following example shows the contents of a database, which contained on `12 jan` one line with `oid` 1023. On `20 feb` this line was split into two parts: 1023 and 1268; see Fig. 4. Finally, the attribute `quality` of one of the lines was changed on `14 apr`. The SQL-queries in Section 4 show how easy it is to produce the update files with new, changed, and deleted objects related to a specific time interval.

[4]Instead of storing the old and new states, it is also possible to store the events only [7, 1]. However, it will not be easy to retrieve the situation at any given point in time.

[5]This is similar to the Postgres model [23]. A temporal SQL extension is described in [22]. In [26] a temporal object database query language for spatial data is presented.

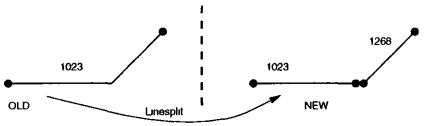


Fig. 4: A 'line' split into 2 parts

```
line
oid  shape     ... quality tmin  tmax
1023 (0,0),(4,0),(6,2)  1 12jan 20feb
1023 (0,0),(4,0)        1 20feb 14apr
1268 (4,0),(6,2)        1 20feb MAX_T
1023 (0,0),(4,0)        2 14apr MAX_T
```

Predecessor and successor
A query producing all historic versions of a given object only needs to specify the `oid` and leave out the time attributes. This does work for simple object changes, but does not work for splits, joins, or more complicated spatial editing. However, this information can always be obtained by using spatial overlap queries with respect to the given object over time, that is, not specifying `tmin/tmax` restrictions.

3 Locking, check-out, and check-in

A GIS is different from many other database applications, because the topological edit operations can be complicated and related to many old and new objects. This results in *long transactions.* During this period other users are not allowed to edit the same theme within this rectangular work area. They must also be allowed to view the last correct state before the editing of the whole database. An alternative to locking is versioning [5], but it is impossible to merge conflicting versions without user intervention. Therefore, the edit locking strategy is used and this is implemented by the table `lock`.

As the database must always be in a consistent state, it may not be polluted with 'temporary' changes that are required during the topological edit operations. This is the motivation for the introduction of a *temporary work copy* for the GIS-edit program; e.g. X-Fingis [10, 11, 13]. The copy is made during *check-out* and is registered in the `lock` table. This is only possible in case no other work areas overlap the requested region with respect to the themes to be edited. The database is brought from one (topologically) consistent state to another consistent state during a *check-in.* It is important that all changes within the same check-in get the same time stamps in `tmin/tmax` (system time as always). This architecture also has the advantage that it enables an easy implementation of a high level 'cancel' operation (rollback).

Locking a work area
What exactly should be locked when a user specifies a rectangular work area? Of course, everything that is completely inside the rectangle must be locked. This is achieved at the *application* level: check-out and check-in. Objects that cross work area boundaries could also be locked, but this may affect a large part of the database. Other users may be surprised to see when they want to check-out a new non-overlapping part (rectangle), this is impossible due to elongated objects that are locked. Therefore, the concept of *partial locks* is introduced for

these objects: the *coordinates* of the line segment crossing the boundary of the work area are not allowed to change. Together with the fact that the rectangular work areas can never overlap, this implies that the other changes to the edges and faces that cross the borders of two work areas are *additional* and can be merged in the database. Therefore these objects do not have to be locked, but have to be checked in with some additional care. It is possible that two check-ins want to modify the same object; see Fig. 5. If no care is taken and both check-ins replace the object, then only the second version is stored and the changes from the first are lost. Therefore, the following steps must be taken for every changed object crossing the work area boundary:

- refetch the object from the database and acquire a *database* update lock for this object;
- if other changes have occurred, then 'merge' these with the work area version of objects;
- reinsert the 'merged' object in database and release the database update lock.

The 'solution' for avoiding deadlocks, is to allow only one check-in at a time (check-in queue). So, all check-ins are processed sequentially.

Errors and improvements

Errors in the past with respect to data collecting or entering pose a difficult problem: should these be corrected by changing the history `tmin/tmax`? Because of possible consistency problems it was decided not to do so. An alternative solution is to mark error objects by setting an additional attribute `error_date`.

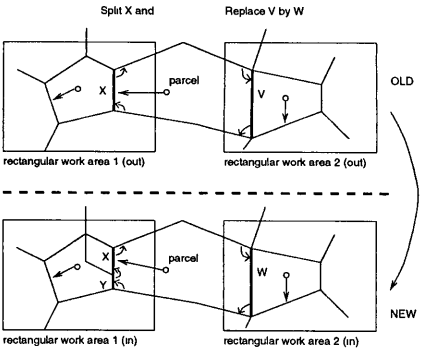


Fig. 5: Difficult check-in rectangular work areas

Another special case is the result of geometric data quality improvement. After obtaining new accurate reference points and 'rubber sheeting' related objects, many relatively small changes occur. It was decided to treat these as normal updates, because the customers must also have the same geometric base as the data provider. Otherwise, potential topological errors may occur (in the future) due to these small differences in the coordinates. However, the customers must be informed about quality improvement, because they will receive large update files.

4 Update files

As explained in the introduction, after an initial full delivery of the data set, the customers receive periodic update files, which contain the differences with respect to the previous delivery [16]. The time interval for a typical update file starts at the begin point in time `t_beg`

and stops at the end point in time t_end. The update files are composed of two parts: OLD *(in Dutch* WAS*):* deleted objects and old versions of changed objects; NEW *(in Dutch* WORDT*):* new objects and new versions of changed objects.

Besides selecting these data from the database (using SQL queries with time stamps), the production of update files at least has to include reformatting the database output in the national data transfer standard NEN-1878 [17] or some other desired data transfer format. The object changes might occur in attributes, such as topological references, which the customer does not receive. These invisible changes can be either filtered out (*signif_changes*) or may be left in the update file (*all_changes*). There are two ways of interpreting the begin (t_beg) and end (t_end) time related to an update file: as a complete time *interval* or as two individual *points* (*instants*) in time. In the second case, the customer is not interested in temporary versions of the objects between the two points in time t_beg and t_end. This results in four different types of update files:

1. interval_all_changes: all changes over time interval (t_beg, t_end] including t_end, with delivery of all temporary object versions.

```
/* deleted/updated objects */
select * from line l where
  t_beg < l.tmax and l.tmax <= t_end;

/* new/updated objects */
select * from line l where
  t_beg < l.tmin and l.tmin <= t_end;
```

In case an object is updated two times, two versions of old objects (OLD: x,t1 and x,t2) and two versions of new objects (NEW: x,t2 and x,MAX_TIME) will be included in the update file; see the example below:

```
                          oid=x,
            oid=x,        tmax=MAX_TIME
oid=x,      tmax=t2     |------------>
tmax=t1   |-----------|
----------|           t2
          t1
t_beg     (time line)       t_end
--0--------------------------X----->
```

2. points_all_changes: only changes comparing the two points in time t_beg and t_end, excluding all temporary versions, have to be delivered. This means that the object versions have to overlap in time either t_beg (deleted/updated objects) or t_end (new/updated objects).

```
/* deleted/updated objects */
select * from line l where
  t_beg < l.tmax and l.tmax <= t_end
                 and l.tmin <= t_beg;

/* new/updated objects */
select * from line l where
  t_beg < l.tmin and l.tmin <= t_end
  and t_end < l.tmax;
```

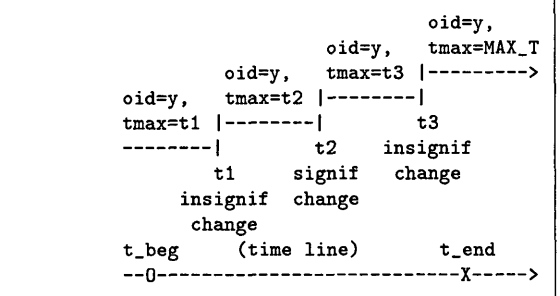
In the example above this will produce only one version of the old object (OLD: x,t1) and only one version of the new object (NEW: x,MAX_TIME).

3. interval_signif_changes: all changes over time interval (t_beg, t_end] with respect to the delivered attributes (A1,A2,...,An) are included in the update file. Ai can be a geometric data type. As the data has to be reformatted anyhow

by the front-end application in order to produce the standard transfer format NEN-1878, it is easy to include the filter for significant changes in this application (especially if the input data is sorted on `oid`):

```
select l.oid,l.tmax,l.A1,l.A2,...
from line l
where /* deleted/updated */
  t_beg < l.tmax and l.tmax <= t_end
  or /* new/updated */
  t_beg < l.tmin and l.tmin <= t_end
sort by l.oid, l.tmax;
```

4. points_signif_changes: all changes comparing the two points in time `t_beg` and `t_end` with respect to the delivered attributes `(A1,A2,...,An)` are included in the update file. It is now not true anymore that the reported object versions have to overlap in time either `t_beg` (deleted/updated objects) or `t_end` (new/updated objects), because they can be related to insignificant changes. It could be that a significant change occurs somewhere in the middle; see the example below:



In general, many insignificant versions of an object, w.r.t. the attributes for a customer, may precede and/or follow a version with a significant change. These should be temporarily glued together with versions related to insignificant changes; not in the database itself. This can be included easily in the application program in two steps: first 'glue', then filter out glued object versions, which do not overlap the two points in time: `t_beg` and `t_end`.

5 Future work

Visualizing changes over time requires implementing specific techniques [2, 12, 14] in a geographic query tool such as GEO++ [25]. The following is an overview of possible techniques to visualize spatial temporal data; more details can be found in [24]. *Double map:* Display besides each other the same region with the same object types but related to two different dates. *Change map:* Display the changed, new and deleted objects over a specified time interval on top of the map. *Temporal symbols:* Use a static map with thematic symbols for a temporal theme; e.g. depicting dates, change rates, order of occurrence, etc. *Space-time aggregation:* Aggregate the (number of) changed, new, and deleted objects to larger units in order to visualize the change rate in different regions. *Time animation:* Visualize changes through an animation by displaying the same region and object types starting at `t_beg` in n steps to `t_end`. *Time as third dimension:* Visualize changes over time, by using the third dimension for time. The user navigates through this 3D-space; see Fig. 6.

Although many aspects of maintaining topology and time in a database have been described, there are still

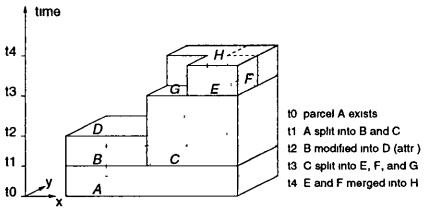


Fig. 6: 3D visualization of parcel changes over time

some open questions: 1. should we try to model the future?, and 2. how long should the history be kept inside the database tables? The current proposal is to keep the information in the database forever.

Returning to the first question: in addition to the history we might also want to model the (plans for the) future. In contrast to the past were there is only one time 'line', the future might consist of alternative time 'lines', each related to a different plan. There is a different type of 'time topology' for these future time lines; see [6]. In this case multiple versions are needed [5].

6 Conclusion

This paper shows how changes in map topology may be recorded in a temporal database by only using the `oid` part of the key for topology references and omitting the time part `tmax`. This avoids updating the neighbors in many cases. The check-in/check-out of workfiles enable long transactions and assure that the database is always in a correct state and that the spatial topology references are always correct. Further, the temporal topology is also correct as object versions are adjacent on the time line. The model allows 1. easy reconstruction of the situation for every given point in time, and 2. easy detection of all changes over a *time interval* or between two *points in time* for the production of several type of update files.

Acknowledgments
Many ideas with respect to storing topology and history were developed in early discussions with Chrit Lemmen. The developers of GEO++ (Tom Vijlbrief) and X-Fingis (Tapio Keisteri and Esa Mononen) were helpful with there comments. Paul Strooper, once again, thoroughly screened this paper, which caused a significant improvement. Finally, several colleagues (Berry van Osch, Harry Uitermark, Martin Salzman, Bart Maessen, Maarten Moolenaar, Peter Jansen, and Marcel van de Lustgraaf) volunteered to act as reviewers and all did give useful suggestions.

References

[1] C. Claramunt abd M. Theériault. Toward semantics for modelling spatio-temporal processes within gis. In *7th SDH*, volume 1, pages 2.27–2.43, August 1996.

[2] C. Armenakis. Mapping of spatio-temporal data in an active cartographic environment. *Geomatica*, 50(4):401–413, 1996.

[3] Bruce G. Baumgart. A polyhedron representation for computer vision. In *National Computer Conference*, pages 589–596, 1975.

[4] CA-OpenIngres. Object Magemenent Extention User's Guide, release 1.1. Technical report, June 1995.

[5] M. E. Easterfield, R. G. Newell, and D. G. Theriault. Version management in gis – applications and techniques. In *EGIS'90*, pages 288–297, April 1990.

[6] A. U. Frank. Qualitative temporal reasoning in GIS – ordered time scales. In *6th SDH*, pages 410–430, September 1994.

[7] C. M. Gold. An event-driven approach to spatio-temporal mapping. *Geomatica*, 50(4):415–424, 1996.

[8] C. S. Jensen, J. Clifford, and R. Elmasri. A consensus glossary of temporal database concepts. *SIGMOD Record*, 23(1):65–86, 1994.

[9] Kadaster, Directie Geodesie. Handboek LKI - extern, technische aspecten. Technical report, Dienst van het Kadaster en de Openbare Registers, November 1989. (In Dutch).

[10] Karttakeskus, Helsinki, Finland. Fingis User Manual, version 3.85. Technical report, 1994.

[11] T. Keisteri. Fingis – software and data manipulation. In *Auto Carto London*, volume 1, pages 69–75, September 1986.

[12] M. J. Kraak and A. M. MacEachren. Visualization of the temporal component of spatial data. In *6th SDH*, pages 391–409, September 1994.

[13] KT-Datacenter Ltd., Riihimäki, Finland. X-Fingis Software V1.1, INGRES version. Technical report, October 1994.

[14] G. Langran. *Time in Geographic Information Systems*. Taylor & Francis, London, 1992.

[15] C. Lemmen and P. van Oosterom. Efficient and automatic production of periodic updates of cadastral maps. In *JEC-GI'95*, pages 137–142, March 1995.

[16] C. H. J. Lemmen and B. Keizer. Levering van mutaties uit de LKI-gegevensbank. *Geodesia*, 35(6):265–269, September 1993. (In Dutch).

[17] NEN-1878. Automatische gegevensverwerking - Uitwisselingsformaat voor gegevens over de aan het aardoppervlak gerelateerde ruimtelijke objecten. Technical report, Nederlands Normalisatie-instituut, Juni 1993. (In Dutch).

[18] D. Peuquet and L. Qian. An integrated database design for temporal gis. In *Proceedings of the 7th International Symposium on Spatial Data Handling, Delft, The Netherlands*, pages 2.1–2.11, August 1996.

[19] D. J. Peuquet and E. Wentz. An approach for time-based analysis of spatio-temporal data. In *6th SDH*, pages 489–504, September 1994.

[20] H. Raafat, Z. Yang, and D. Gauthier. Relational spatial topologies for historical geographical information. *IJGIS*, 8(2):163–173, 1994.

[21] A. A. Roshannejad. *The Management of Spatio-Temporal Data in a National Geographic Information System*. PhD thesis, Enschede, The Netherlands, Twente Univeristy, 1996.

[22] R. T. Snodgrass, I. Ahn, and G. Ariav. Tsql2 language specification. *SIGMOD Record*, 23(1):65–86, 1994.

[23] M. Stonebraker and L. A. Rowe. The design of Postgres. *ACM SIGMOD*, 15(2):340–355, 1986.

[24] P. van Oosterom and B. Maessen. Geographic query tool. In *JEC-GI'97*, page ?, April 1997. To be published.

[25] T. Vijlbrief and P. van Oosterom. The GEO++ system: An extensible GIS. In *5th SDH*, pages 40–50, August 1992.

[26] A. Voigtmann, L. Becker, and K. H. Hinrichs. Temporal extensions for an object-oriented geo-data-model. In *7th SDH*, volume 2, pages 11A.25–11A.41, August 1996.

[27] M. F. Worboys. Unifying the spatial and temporal components of geographical information. In *6th SDH*, pages 505–517, September 1994.

SIMPLE TOPOLOGY GENERATION FROM SCANNED MAPS.

Dr. Christopher Gold, Geomatics Research Centre,
Laval University, Quebec City, Qc, Canada G1K 7P4
(418)656-3308, Christopher.Gold@scg.ulaval.ca

ABSTRACT

For many GIS applications the data entry component is the most expensive, and frequently makes the difference between success and failure of the project, in terms of both time and money. The most time-consuming part of manual digitizing is usually the generation of correct topology, which usually involves many error correction steps. The automatic processing of scanned maps appears to have many advantages, but in practice the line extraction process does not always produce good topology automatically.

The situation with scanned maps may be improved with a conceptual change of emphasis. Instead of concentrating on the black pixels (linework) one may emphasize the white pixels (polygonal areas) and build the relationships from these. The process has three steps. Firstly a set of white pixels are selected, at a user-specified distance from the black linework, and these are given a polygon label on the basis of a flood-fill algorithm that scans all connected white pixels. Secondly these selected pixels are used as data points to generate the standard Euclidean point Voronoi diagram. Thirdly, this structure is scanned to extract only those Voronoi boundaries between pixels having different polygon labels. The result of this operation is a set of vector chains or arcs that are guaranteed to separate regions of white space, and are guaranteed to connect precisely at nodes.

Experiments were made on two types of map: a simple urban cadastral map, with lot boundaries and building outlines; and a typical Quebec forest map sheet that had been retraced by hand to preserve the forest stands and eliminate all the other superimposed data types. The cadastral map showed excellent automatic topology generation, and accurate linework. Black linework or symbols that are unclosed are not preserved with this method. Unclosed polygons are also lost, and may need pretreatment. In the case of the full forest map, containing about 3000 polygons, manual editing might be required if stand boundaries are too close together - for example along river valleys. Processing time on a small Sparcstation 10 was less than 30 minutes at maximum resolution. The most difficult part of the process was georeferencing the scanned image to the paper map or ground coordinates. A byproduct of the process was the automatic detection of polygon centroids - in this case defined as the centre of the largest circle falling in the polygon. The results were imported directly into the vector GIS without any cleanup or intersection- testing operations. In many cases the approach described here may make significant savings in the map entry process.

INTRODUCTION

The use of maps for planning and analysis depends heavily on their availability in digital form within the computer. This is usually a major bottleneck. In particular, the demands of a GIS include the topological structuring of map linework to form networks and polygons. This requires careful manual digitizing in order to produce a suitably structured map which is then usable for analysis. This is a long and error-prone operation, and is usually the major cost. This bottleneck in map input holds true for any application using a vector GIS. The underlying problem is the difficulty of taking more-or-less precise coordinates and converting them into a structured graph representation within the computer. This is difficult for a variety of reasons - the imprecision of coordinates and the traditional use of a line-intersection model of space, among others - but one of the basic issues is the formal definition (in advance) of what are the classes of objects being detected during the input process. Our experience has shown that, if this is clearly understood, the combination of appropriate object labelling with a general-purpose implementation of a spatial model may permit relatively simple data input. In particular, one should be clear as to whether one is attempting to define line (arc) or area (polygon) features.

Some work has been undertaken to attempt to reduce this bottleneck. Gold et al. (1996) worked on the problem of improving the speed of forest map digitizing by emphasizing the specification of the polygons themselves, rather than the bounding arcs or chains. This has been shown to improve map preparation times significantly at the operational level. It was, nevertheless, a manual method.

An alternative approach is to scan the map, and then to process the resulting (black and white) image. Various commercial products exist for the "skeletization" or thinning of the pixels forming a line. These approaches have, however, run into difficulties with the extraction of good topology - it is difficult to produce a satisfactory vector skeleton that forms a complete set of polygon boundaries. Nevertheless, an automated technique would be a great help in map input.

Much experience has shown that the traditional data input model for GIS is cumbersome and error-prone. This approach could be called the "line-intersection" model of space, since the manual digitizing method involves entering individual chains of x-y coordinates, and the computer program first searches for intersections between these chains. Once intersections are found the system attempts to construct a graph model of the desired map - most often a polygon map. Many errors are possible at this stage, usually related to missing or duplicate intersections, causing difficulties in identifying nodes and completed polygons. Thus all spatial relationships are based on detecting intersections of lines or chains.

ALTERNATIVES TO THE LINE-INTERSECTION MODEL

There are alternatives to the line-intersection model of space. The most obvious is the raster or grid model. Here there are no particular map objects, but only an attribute associated with a particular square tile. This has the advantage that neighbour relationships are implicit in the whole structure (the tiles to the north, south, east and west), but it is rather awkward for the identification of specific map objects, such as roads or polygons. A third approach, which has been used with some success in recent years, is to combine the advantages of both systems: a set of map objects (as in a vector GIS) with a single associated tile (rather like a raster cell). This would give a set of spatial adjacency relationships between adjacent tiles - and hence between each tile's generating object. While various definitions of these tiles may be possible, the most obvious is the proximal definition used to generate the Voronoi diagram. Here each tile or cell contains all spatial locations closer to the generating object (traditionally a data point) than to any other object.

Various algorithms exist for generating this spatial data structure (or its dual, the Delaunay triangulation) for static sets of data points. Examples include Green and Sibson (1978), Guibas and Stolfi (1985), Sugihara and Iri(1989) and Lawson (1977). Extensions exist for constrained triangulations (Lee and Lin, 1986), generators that may be points or line segments (Lee and Drysdale, 1981, Fortune, 1987) and dynamic systems where objects may be added, deleted, or moved (Roos, 1993, Gold, 1991). What is of particular interest here, though, is that the algorithms give a form of automatic topology (Gold, 1994). Aurenhammer, (1991) gives a good review. Even with the algorithms for the static Voronoi diagrams of points, which will be adequate for this paper, the "topological" structure is built automatically, with reasonable levels of robustness. It is an attractive idea to attempt to take this property and to apply it to various GIS data entry problems.

AUTOMATIC TOPOLOGY GENERATION - MANUAL DIGITIZING

In Gold et al. (1996) the forest map problem was attacked by focusing on the primary objects of interest - the forest stands, rather than the boundaries. One or more generating points were digitized within each stand, and given the stand label. It was hoped that the cells would approximate the extent of each stand boundary. Further experimentation showed that the best approximation to the stand occurred when points were digitized closely around the interior of each polygon (Fig. 1a), the Voronoi cells generated (Fig. 1b) and then boundaries between those cells having the same label were suppressed. This gave a good approximation to the boundaries between stands.

The dashed lines are the original boundaries in Fig. 1c and the solid lines are the boundaries extracted from the Voronoi diagram. This digitizing was done

Fig. 1. a) Points digitized around the interior of each polygon;
b) the Voronoi cells generated;
c) the polygon boundaries generated (solid lines), compared with the original map (dashed lines).

manually, as the operator was able to distinguish between stand boundaries and the other information on the paper map. The approach was much more intuitive, as the operator focused on the objects of interest - the sequentially-labelled polygons - and not on the boundaries themselves. Operators could be trained rapidly, and digitizing time greatly reduced. The boundaries thus produced had errors well within the limits of the photo-interpretation used to generate the original paper maps.

The algorithm used was based on an early visibility-ordering approach for triangulations (Gold and Maydell, 1978). This guaranteed that triangles would be processed in a front-to-back order. This was modified to draw the Voronoi cell boundaries, suppressing those boundaries between vertices having the same label. Linked-lists were maintained, as in Gold and Cormack(1987), to preserve all complete arcs between triple-junctions (nodes). A single pass through the triangulation, with no searching, was all that was required to extract the complete polygon-arc-node topology for direct entry into a traditional vector GIS. Valid polygon centroids were generated automatically. It was guaranteed that the coordinates of arc end-points matched each other to form nodes, and that the result would always be a valid topological structure. Operator errors were confined to mis-labelling items or forgetting to digitize the interior of some polygon, and these could easily be detected, corrected, and the map re-built. For further details see Gold et al. (1996).

AUTOMATIC TOPOLOGY GENERATION - SCANNED MAPS

The next question concerned the possibility of eliminating the manual digitizing process entirely while preserving the automatic topology generation. The manual process was simulated by the use of functions similar to mathematical morphology (Serra, 1982). In mathematical morphology, binary images are processed using two operators: erosion (shrinkage) and dilation (expansion). The objective of the experiment was to take scanned polygon maps (either forest or urban) and use image processing techniques to generate a fringe of points around the black pixels. These points would then be entered into the Voronoi diagram and have the relevant boundaries extracted as in the above manual digitizing case.

URBAN MAPPING - SYMBOL EXTRACTION

In the case of urban mapping, perhaps the most interesting work is that done by Burge and Monagan (1995), as it is similar to the forest mapping project mentioned above in that it is based on Voronoi diagrams. It differs in that they have been developing a method for extracting features from scanned cadastral maps, with emphasis on the extraction of symbols, dashed lines and character strings. In summary, their procedure generates labelled points associated with connected sets of "black" pixels. These are inserted into the Voronoi diagram, and edges are removed between points with the same label, giving an "area Voronoi

diagram" with a cell around each image element. Based on this, sets of dots, dashes or symbols are grouped together. While not discussed much in their papers, the pixels forming the linework of the cadastral maps are also grouped to form line image elements, but no attempt was made to structure the lot boundaries, for example, in the sense of GIS topology - indeed, all connected polygon boundaries will have the same label. The authors suggest that the identification and removal of symbol groups will facilitate the vectorization of the linework by other algorithms.

SCANNED MAPS - TOPOLOGY EXTRACTION

The work described in this paper attempts to concentrate on linework topology, based on the approach previously used on forest maps. Like the work of Burge and Monagan, labelled points are inserted into the simple Euclidean point Voronoi diagram. Again, boundaries are extracted from this diagram and saved only if they are between points with different labels (using the algorithms developed in Gold and Maydell, (1978), Gold and Cormack, (1987) and Gold et al., (1996)). In both projects the Voronoi diagram (or Delaunay triangulation) construction may be performed on O(n log n) time, and boundary extraction in O(n) time - although the boundary extraction algorithm of Gold et al. appears to be simpler. Unlike Burge and Monagan, however, the symbols themselves were not of interest. The primary bottleneck in GIS is the extraction of "topology" from manually digitized or scanned maps, and the emphasis on collecting all individual arcs forming the polygon boundaries and then connecting them together imposes a heavy workload on computer and operator. Thus our main interest was to extract complete polygons rather than symbols.

The forest mapping project, with its processing of digitized points inside each polygon, led to an emphasis on polygon detection rather than the identification of connected "black" pixels once we started processing scanned maps. In the manual digitizing project, points at the edge of each polygon were given the polygon label, and the Voronoi and boundary extraction functions selected arcs between differently labelled polygon points. An obvious extension was to attempt to process scanned maps in the same fashion, thus removing the need for manual digitizing. A set of "fringe" points were generated at a distance D from any black pixels in the scanned image, and then thinned to be a distance S apart. In practice we were able to set D to one or two pixels, and S to the same value. A standard flood-fill algorithm was then used to assign the same polygon label to all fringe points within the same connected white-space region.

These label points are then entered into the Voronoi diagram and boundary extraction modules, which generate the Voronoi diagram in Euclidean (as opposed to raster) metric. By the nature of the Voronoi diagram, the polygon topology is always complete, because all points with the same label whose Voronoi cells are connected will have a boundary around them. Any black pixels that do

not connect to enclose a white region will have a fringe of label points generated, but the flood-fill algorithm, operating within each connected white region, will give them all the same label. Consequently none of their Voronoi boundaries will be extracted. Thus, using the Voronoi boundary extraction procedure in this case identifies polygonal regions, while the approach of Burge and Monagan based on connected black pixels identifies isolated symbols. In addition, the system can estimate a good centroid position for each polygon. These are calculated as the centre of the largest circumcircle of any triangle having all three vertices within the same polygon.

Fig. 2a shows a small scanned urban map, containing lot boundaries, buildings, labels, road boundaries and point symbols. The fringe label points were added to the image as described above. Fig. 2c shows the fringe label points in the southernmost portion of Fig. 2a. The Voronoi diagram was then generated, and the boundaries between differently-labelled points were extracted and exported as complete arcs. The results were imported directly into Arc/Info, and as no "clean" operation was required (because all arc ends matched precisely), the map could be viewed directly. The processing is fairly rapid, and the method is simple to implement. Fig. 2b is the final result of simple clean-up operations to remove small closed loops, (causing an error at the corner of one house which had text superimposed), and the use of the Douglas-Peucker algorithm to reduce the number of line segments for straight line boundaries. Notice that the northeastern-most lot boundary is lost, due to a small gap in the drawn boundary. (Nodes are represented by small dots.)

FOREST MAPPING

Fig. 3 shows a detail of the Quebec forest stand map generated by the same method, at 25% of the original scale. The original input was retraced to eliminate roads, contour lines, etc. that had been superimposed on the same map. Processing time for the full map was under 30 minutes on a small Sparcstation, approximately equally split between the image analysis step and the Voronoi plus edge extraction step. The map consisted of approximately 3000 polygons, and the resulting Voronoi diagram had about 500,000 points. Precision of the generated centreline is estimated to be within about one pixel. The algorithm gives a slight displacement of nodes when one arc meets another at right angles, and some editing is required where two input boundaries are extremely close or touching, causing the extracted boundaries to merge in places. The most difficult part of the process was georeferencing the scanned image to the original map or ground coordinates. The arcs and the automatically generated polygon centroids may be directly input int Arc/Info or any equivalent GIS without any need for further processing. Indeed, most forms of topological structure could be extracted directly from the Voronoi diagram as required. Polygon labels may be added interactively to the centroids (although industrial experience shows that this is a slow operation, and the manual data entry described above deserves serious consideration).

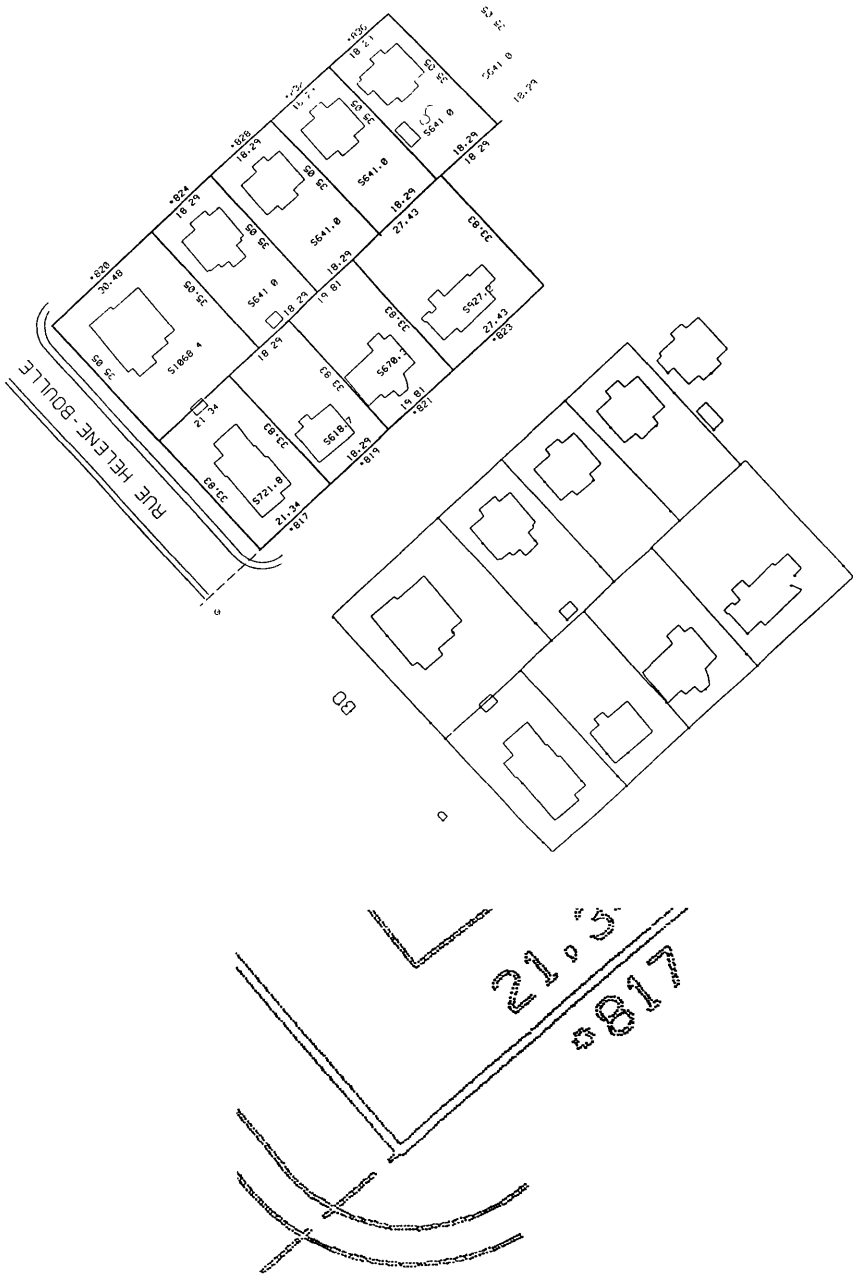


Fig. 2. a) A small cadastral map;
b) the final cleaned-up map.
c) a few of the fringe points generated in the south of the map.

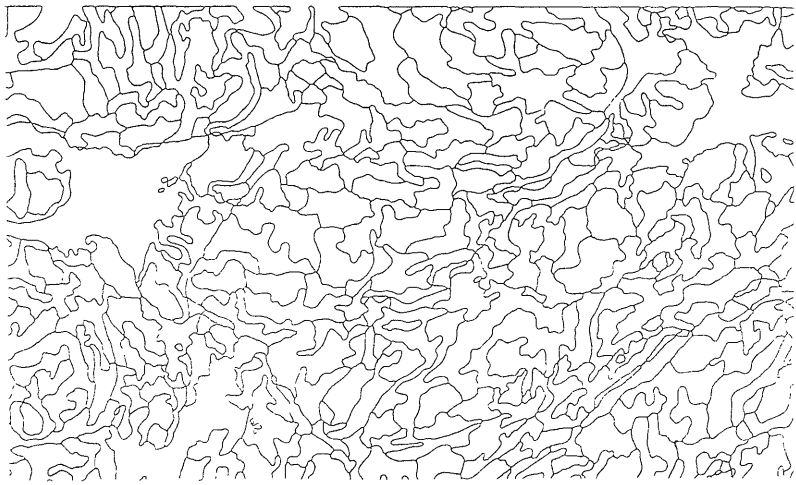


Fig. 3 Results for a portion of a Quebec forest map (at 25% scale), showing the complete polygon topology.

FUTURE WORK AND CONCLUSIONS

The automatic detection of topology within the Voronoi module opens up a variety of other developments. Further information can be extracted from the Voronoi diagram itself, for example which polygons are adjacent to, or enclosed within, other polygons. In addition, Gold et al. (1996) show that almost any GIS topological structure may be extracted from the Voronoi diagram as required. The links between mathematical morphology and the Voronoi diagram are close - dilation functions to form raster buffer zones have their equivalent in the Euclidean Voronoi diagram (Gold, 1991). Vincent (1988) has shown that mathematical morphology operations may be performed directly on the Voronoi cells, not just on raster images. These relationships need to be explored further. Even at this preliminary stage, it is clear that the Voronoi approach to the processing of scanned maps has definite advantages.

ACKNOWLEDGMENTS

This research was made possible by the foundation of an Industrial Research Chair in Geomatics at Laval University, jointly funded by the Natural Sciences and Engineering Research Council of Canada and the Association de l'Industrie Forestiere du Quebec. The author would like to gratefully acknowledge the assistance of Luc Dubois, Weiping Yang, Francois Anton and Darka Mioc with various aspects of the programming and testing.

REFERENCES

Aurenhammer, F., 1991. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Computing Surveys* v. 23, pp. 345-405.

Burge, M. and G. Monagan, 1995. Extracting words and multi-part symbols in graphics-rich documents. In: *Eighth International Conference on Image Analysis and Processing: ICIAP. Lecture Notes in Computing Science, IAPR*, Springer Verlag, 1995.

Fortune, S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, v. 2, pp. 153-174.

Gold, C.M., 1991. Problems with handling spatial data - the Voronoi approach. *CISM Journal* v. 45, pp. 65-80.

Gold, C.M., 1994. Three approaches to automated topology, and how computational geometry helps. *Proceedings, Sixth International Symposium on Spatial Data Handling*, Edinburgh, pp. 145-158.

Gold, C.M. and S. Cormack, 1987. Spatially ordered networks and topographic reconstructions. *International Journal of Geographical Information Systems*, v. 1, no. 2, pp. 137-148.

Gold, C.M. and U.M. Maydell, 1978. Triangulation and spatial ordering in computer cartography. *Proceedings, Canadian Cartographic Association Third Annual Meeting*, Vancouver, pp. 69-81.

Gold, C.M., J. Nantel and W. Yang, 1996. Outside-in: an alternative approach to forest map digitizing. *International Journal of Geographical Information Systems* v. 10, pp. 291-310.

Green, P.J. and R. Sibson, 1978. Computing Dirichlet tessellations in the plane. *Computer Journal*, v. 21, pp. 168-173.

Guibas, L. and J. Stolfi, 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, v. 4, pp. 74-123.

Lawson, C.L., 1977. Software for C-1 surface interpolation. In: J. Rice (ed.), *Mathematical Software III*. Academic Press, New York, pp. 161-194.

Lee, D.T. and R.L. Drysdale III, 1981. Generalizations of Voronoi diagrams in the plane. *SIAM Journal of Computing*, v. 10, pp. 73-87.

Lee, D.T. and A.K. Lin, 1986. Generalized Delaunay triangulations for planar graphs. *Discrete and Computational Geometry*, v. 1, pp. 201-217.

Roos, T., 1993. Voronoi diagrams over dynamic scenes. *Discrete Applied Mathematics*, v. 43, pp. 243-259.

Serra, J., 1982. *Image analysis and Mathematical Morphology*. Academic Press, London.

Sugihara, K. and M. Iri, 1989. Two design principles of geometric algorithms in finite-precision arithmetic. *Applied Mathematical Letters*, v. 2, pp.203-206.

Vincent, L., 1988. Mathematical Morphology on graphs. *Proceedings of SPIE -The International Society for Optical Engineering*, 1988, pp. 95-105.

NEW MAP PROJECTION PARADIGMS: Bresenham Poly-Azimuthal Fly-Through Projections, Oblique Mercator Triptiks, and Dynamic Cartograms

Alan Saalfeld
Department of Civil and Environmental Engineering and Geodetic Science
The Ohio State University
Columbus, Ohio 43210-1275 USA

Abstract

We examine map projections and their distortions in a discretized, time-dependent computer mapping environment; and we propose some new map projection paradigms. The computer environment permits us to display animated projection evolution (realized as a movie of continuous deformation from a perspective view of the original datum surface to the projection surface). An animated projection evolution technique may also be used to produce varivalent projections (cartograms) built by iterative discrete distortion techniques. The discretized environment also allows us to quickly change the viewpoint and the projection orientation (by means of pixel shift operations) to produce a sequence of overlapping maps, each of which is distortion-free (up to sub-pixel resolution) with respect to a moving central point. We also examine methods for producing large scale route strip map sets such that each route segment is distortion-free throughout the strip map in which it is featured.

INTRODUCTION

What would the savvy map user ask for in a map projection these days if he or she knew about the latest possibilities for computer generated maps? Probably the same thing that a hopelessly naive user might request–a totally distortion-free map! While a distortion-free map is and always will be a mathematical impossibility for any region containing 4 or more non-coplanar points, one may, nonetheless, (and for sufficiently large scales) hold displacement distortion to subpixel size and draw a map with no discernible distortion. The following are technology-inspired tactics to (1) minimize perceptible distortion at and around a rapidly moving viewpoint, to (2) minimize distortion along a particular route, and to (3) minimize distortion under a magnifying glass that we baby-boomers are finding increasingly necessary to use to read our maps.

BRESENHAM-TYPE METHODS

Every computer graphics student learns early about J. E. Bresenham's elegantly simple algorithms for coloring pixels one by one to generate raster representations of straight lines [Bre65] and circles [Bre77]. The algorithms

employ elementary integer arithmetic operations of addition and subtraction (and nothing else!) They are fast, robust, and surprisingly easy to implement and to prove correct. This paper (and an associated computer demonstration) attempt to extend the flavor, if not the theory, of Bresenham's work to two and three dimensions by incrementally updating the pixels of the 2-D map of the ground below Dr. Bresenham as he moves along and above the surface of the earth in nice pixel-sized increments. What we accomplish with our incremental methods are real-time azimuthal projections continuously centered directly below a moving observer. There is never any linear or angular distortion at the current viewpoint (always the center of the map). Distortion is always radial (so directions from the current viewpoint are always correct); and concentric circles about the viewpoint delineate "contours of equal distortion." We discuss the different incremental procedures needed to update different azimuthal projections; and we define simple incremental procedures and analyze their resulting projection properties. The central symmetry of all of our adjustments permits shortcut computations (similar to Bresenham's observation [Bre77] that for circles, one needs only compute an arc that is 1/8 the circumference, then reflect in various axes).

The speed of the incremental computations permits a discrete image to be generated at a much higher resolution than the display. The computed higher resolution grid may then be smoothed with a filter (in much the same way that anti-aliasing is often applied to remove the jaggies from Bresenham's line). Moreover, the incremental changes of the observer's movement may be everywhere realized as pixel shifts followed by smoothing or averaging (averaging is necessary when the finer grid pixels only move a fractional amount of the larger pixel size). The observer's movement may be decomposed into its X, Y, and Z components; and the effect on the image of each step in one of the three perpendicular directions may be computed once and stored in a lookup table. Since movements in the three independent directions "almost" commute with each other, we may sum the pixel shift effects of the X, Y, and Z components in any order to determine the net effect.

A Moving Pixel's Perspective

Motion of the viewer translates into apparent motion of an image in the opposite direction. As a train passenger looks out the window; he sees the scenery moving past him in the opposite direction to the train's motion. Objects that are close to the train appear to move more rapidly than distant objects. The moon appears to be keeping up with the train because its relative motion is so slight that it seems to not move backwards at all! The geometry of this apparent motion with respect to the train window is straightforward—the rate that a stationary object appears to move past the window is inversely proportional to its distance from the viewer. One may

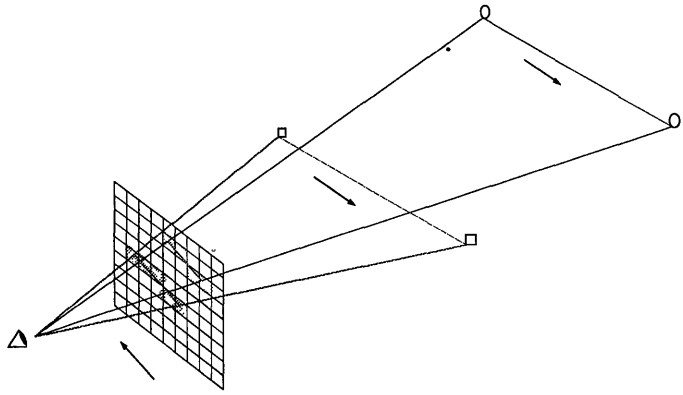


Figure 1: Linear point motion as captured in the digitized image plane.

discretize this apparent motion by imagining that the window is screened (a fine-grained rectangular grid), the viewer's eye is fixed with respect to its distance and position relative to the screen, and the viewed objects appear to move from one tiny rectangle of the screen to an adjacent rectangle as the train moves forward. For the train example, all actual motion and all apparent motion is horizontal! If the vehicle could move vertically as well, but continued in a straight line, then the distant point object would appear to move from cell to cell in the gridded screen in exactly the same pattern as the incremental linear Bresenham algorithm generates successive grid cells, as is illustrated in Figure 1.

Pixel trajectories

Consider now the sequence of screen rectangles (pixels) in which a particular distant point light source appears over time. At any time the light point representation on the screen has a screen position and a screen velocity. If the screen contains a line parallel to the direction of motion of the train, then the lighted pixel's velocity will be constant and linear; hence, the individual pixel's trajectory over time will be correctly modeled by the Bresenham algorithm for painting successive pixels along a line at regularly spaced time intervals. If one knows all of the pixels' velocities (speed and direction) at each instant, then one may integrate the velocities to obtain tracks or trajectories for each individual pixel component of a map's images. The significant notions that we can exploit here are (1) the pixel content (is it black or white or colored?) makes absolutely no difference to the trajectory determination; and (2) the pixel movements repeat their patterns (so that the full image of pixel shifts may be saved and re-used so that they may be applied repeatedly to generate successive images). The movement of objects in the foreground will appear to outpace the movement of more

distant objects; hence, foreground objects may overtake and temporarily obstruct the view of more distant objects, and then the foreground objects will again uncover or reveal distant objects as the foreground objects appear to move past the background objects. If all of the objects are in a single plane parallel to the screen (i.e., all the same distance from the plane of the screen), then the apparent pixel motion on the screen will be completely uniform (same direction and same pixel speed) everywhere. This model is a bit too simplified for our applications!

Figure 2: Pixels' displacement simulates rotational motion.

Let's look at some image updates for which pixel movement is not uniform. Consider the following simplified animation of the spinning earth icon or applet: shaded pixels are displayed in a circle in a way that produces the illusion of a spinning globe, as illustrated in Figure 2. The pixel shifts that accomplish this illusion produce trajectories along the perspective projection of parallel circles of latitude. The speed of the pixel movements is greater at the equator and diminishes near the poles. The speed near the edges of the circular disk also diminishes to produce the effect of less motion in the viewing screen plane (the greater component of the rotational motion is perpendicular to the viewing screen plane). Note that the persistence of the specular reflection (a lightening of pixels as they approach the center of the circle) reinforces the effect that the globe itself is rotating and the viewer and light source remain fixed. It is worthwhile to emphasize that the pixel shifts relative to their reference position in the circular display are identical from the first image to the second, from the second to the third, and from the third to the fourth. Each successive image represents a 20° rotation; and the pixel shifts are completely determined by that fact (and not by whatever happens to occupy a pixel location at any moment).

If the globe were transparent, and if we were far enough away from it (so that all rays that we perceive are effectively parallel), we would see each point on the earth trace out an ellipse. The pixel motion necessary to trace out an ellipse is easily described in terms of Bresenham's circle drawing routine: if we give our pixels an aspect ratio of b/a, then the figure that we draw with Bresenham's circle drawing routine is actually an ellipse; and the flattening of that ellipse is precisely $(a - b)/a$.

We may offer one final illustration based on the appearance of a rotating spherical globe as viewed from space: Suppose that we position ourselves one earth diameter from the earth's surface and we place a giant convex lens

tangent to the earth at the point nearest to us, as illustrated in Figure 3. If the lens bends the light as shown in Figure 3, with viewer and antipodal point as conjugate points, then the viewer "sees" the stereographic projection. If the viewer imagines a fine pixel grid superimposed on the central plane of the lens, then lateral movement of the viewer coupled with corresponding movement of the lens will correspond to sliding the tangent point for the plane of the azimuthal stereographic projection.

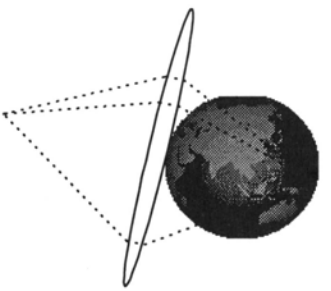


Figure 3: Optical construction of stereographic projection.

The relative shifts of pixels corresponding to the same point on the earth's surface is easy to describe and compute. The stereographic projection is conformal; and the radial displacement on the projection of a point at distance $R\theta$ along a great circle arc from the point of tangency is $2R\tan(\theta/2)$. The scale factor at distance $R\theta$ along a great circle arc from the point of tangency is $2/(1+\cos\theta)$. The scale factor is the relative size of velocity vectors corresponding to movement at the point in question. It is precisely those velocity vectors that provide the magnitude of the pixel shifts for the viewpoint change on the projection surface.

Graphics speed-up tricks

If observer's movements proceed linearly (in a constant direction), then double-buffering techniques can be applied easily to "leap-frog" the generated images. The computation (and buffer loading) of all "odd" observer positions may proceed apace with the computation (and buffer loading) of all "even" positions so that the dual buffers may stay synchronized as they flush alternate frames to the screen. This and other "fast drawing" tricks of computer graphics theory that may facilitate real-time map updates involving continuously changing projections [Pet95]. The key to updating projections with a moving viewpoint is that if the relative motion necessary to update the viewpoint does not change, then the pixel movements (in terms of pixel locations) are identical. Only the pixel content changes from shift to shift. In a manner identical to constructing and repeating the shifts of pixels necessary to simulate earth rotation, we may reposition

the tangent point of our azimuthal projection by shifting all pixels in the opposite direction. To move right, we take the pixel on the right and shift it left. A pixel that is far away will have to shift a greater distance. For example, a pixel that is 90° away from the tangent pixel will move twice as fast because the scale factor is exactly two in the projection at 90° from the point of tangency.

CONFORMAL TRIPTIKS

In this section we examine some opportunities to construct and use maps that are conformal along a route

Oblique Mercator Strip Maps

An oblique Mercator projection of the sphere, with its cylinder tangent to the great circle joining two points of interest, provides a distortion-free representation of all points along the shortest path between those two points [Sny94]. Such a representation provides a pilot or a navigator with a direct routing from start to destination. Any point along the great circle route corresponds to a point of no distortion on the map. For ground-following routes, one may approximate the multi-directional path by a sequence of great circle arcs on a sphere modeling the Earth; then one may compute oblique Mercator projections along each great circle arc. One might present each arc's projection separately; or one might even "blend" the projections using other available computer graphics techniques [Far93]. We will visit blending once again when we briefly touch upon homotopy and homotopic projections.

Minimizing distortion along and near a closed path

One may minimize distortion along a path by keeping the function conformal in a neighborhood of the path and also maintaining a constant scale along the path. Since the length of a path on the map differs from the length on the datum surface by a factor of scale (which is constant), we must have that relative lengths along partial paths are preserved everywhere. Chebyschev had conjectured (and others later proved) [BS95] that a conformal scale-constant mapping of the closed boundary of a simply connected region to the closed boundary of another simply connected region extends conformally to the interior of the regions in a way that minimizes scale variation within the region.

Cheng's Conformal Polyconics

Yang Cheng [Che92] showed how to attach a tangent developable surface to *any* smooth rectifiable curve on a datum surface (sphere or ellipsoid); and from that construction, he is able to extend the projection of that curve

conformally to a neighborhood of that curve with no distortion along the tangent curve itself. Cheng's methods have been applied in detail to specific important curves such as satellite ground tracks [Che96]. They may also be applied readily for any route on the sphere or ellipsoid for which we may compute geodesic curvature. We must merely construct a plane curve whose curvature matches the geodesic curvature of the curve on the sphere or ellipsoid; then we may widen, expand, or buffer the curve in the plane to produce a swath (or wiggly triptik!) on which we may construct a conformal mapping of a neighborhood of the curve. This conformal mapping will have no scale distortion along the curve itself.

DYNAMIC CARTOGRAMS

Morphing technology in computer graphics has created a standard toolbox for animators, graphic artists, and image processors. We focus here on describing a subset of morphing tools that possess desirable map projection properties such as conformality and equivalence; and we show how those tools can be applied effectively to create an interesting collection of map products.

Homotopies

A homotopy may be regarded as a continuous deformation over time of one function to another. Formally, if $f : X \to Y$ and $g : X \to Y$ are two functions on the same domain X and range Y, then we say that f and g are homotopic if there exists a continuous function $\phi : X \times [0,1] \to Y$ such that for all $x \in X$, $f(x) = \phi(x,0)$ and $g(x) = \phi(x,1)$. One may regard the second parameter of the bivariate function ϕ as a time parameter: at $t = 0$ the function ϕ behaves like f; at $t = 1$ the function ϕ behaves like g; and for $0 < t < 1$, the function ϕ changes continuously with respect to t.

Because each cross-section $\phi : X \times \{t_0\} \to Y$ of a homotopy is only required to be continuous (and not necessarily bijective), the intermediate slices of two homotopic projections f and g may not be projections in the usual sense (because of collapsing). For example, if one rotates any projection through 180°:

$$\text{If } f(x) = (\rho, \theta), \qquad \text{then } g(x) = (\rho, \theta + 180°),$$

then defines a straight line deformation from f to g:

$$\phi(x,t) = tg(x) + (1-t)f(x),$$

then the function ϕ collapses to the origin everywhere at $t = 1/2$.

Often the function ϕ is called the convex combination of g and f. If g and f are analytic complex-valued functions of a complex variable, then every convex combination of g and f will also be an analytic function. Analytic is

the same as conformal or angle preserving provided the function does not collapse somewhere. There are many other possibilities for guaranteeing conformality of combined functions. All complex arithmetic operations return conformal functions.

Complex Variables and Conformal Functions

Conformal functions have some amazing properties related to the structure of the complex number field that mathematicians have discovered and studied. These properties are at the same time very constraining and yet very powerful in nature. One very important property is that the complex numbers are not simply two-vectors, they possess an algebraic interaction that manifests itself very nicely in the geometry. To multiply two complex numbers by adding their angles and multiplying their magnitudes is both incredible and liberating. Another defining property is differentiability in the complex variable sense. A derivative exists at each point; and it may be computed as a limit from any direction. A directional derivative is a scale factor in the particular direction. For conformal functions, all directional derivatives at a single point are the same (complex) value. In other words, at each individual point in the domain, the scale factor (magnitude change of any tangent vector) in every direction is the same; and so is the rotation component of each tangent vector. Here are some of the other amazing properties of conformal functions [Cur43]:

1. If a function is once (complex) differentiable in an open neighborhood of a point, then it is infinitely differentiable in the neighborhood of the point.

2. Each conformal mapping is fully determined in a maximum circular region about any point by the first, second, third, and higher order derivatives at the single point.

3. A conformal function is fully determined in a maximum circular region by its values on any open set, however small. (This is perhaps the most constraining property since we lose all freedom to assign our own set of values to a conformal function even far from the defining site.)

4. A conformal function has a power series expansion in a complex variable. The radius of convergence extends exactly as far as the nearest singularity of the function.

5. We have some bad news, too: we want to stay away from singularities. In any neighborhood of a singularity, a conformal function assumes every possible sufficiently large value.

6. For any simply connected region, there exists a conformal function that sends the unit disk onto the region. (This says that we can

preserve local scale and shape and still distort the global picture as much as we want. This seems counter-intuitive!)

7. A conformal function is an open map. It sends open sets to open sets.
8. Any two conformal functions that agree on an infinite set of points agree everywhere.
9. The composition of two conformal functions remains conformal.
10. Boundary conditions may preclude the existence of any satisfying conformal functions.
11. Homotopy and conformality meet [ST83] in Cauchy's Theorem: If a closed path $\gamma(t)$ is homotopic to the null path in a region of differentiability of f, then $\int_\gamma f = 0$. If two closed paths $\gamma_1(t)$ and $\gamma_2(t)$ are homotopic in a region of differentiability of f, then $\int_{\gamma_1} f = \int_{\gamma_2} f$.
12. The homotopy properties guarantee the existence of anti-derivatives as well as infinitely many derivatives.

Quasiconformal Functions

Quasiconformal functions [Ahl87] are as close to conformal as one may get when boundary conditions are such that conformality is impossible. If we use the eccentricity of the ellipse of the Tissot indicatrix [Las89] to measure our failure to achieve conformality, then quasiconformal functions have the smallest eccentricity possible while still satisfying the defining boundary conditions. Some important quasiconformal functions correspond to conformal transformations followed by affine transformations (which wind up flattening all Tissot ellipses in the same direction and by the same fractional amount).

Area Preservation and Area Distortion

Waldo Tobler [Tob86] and Lev Bugayevskiy [BS95] have studied the differential equations of varivalent transformations; and Tobler has produced several programs to implement his methods [Tob74]. An opportunity to examine the discretized version of the transformations exists for us to apply the incremental methods described in this paper to the theory of Tobler and others.

FINAL REMARKS

We have only had time and space to present what appears to be a laundry list of possibilities for new map projection paradigms. We certainly do not claim to have exhausted the possibilities; and our limited perspective is just that—quite limited. Nevertheless, we believe that we have highlighted a

series of related opportunities; and we hope that our viewpoint stimulates new research into map projections and their uses.

References

[Ahl87] Lars V. Ahlfors. *Lectures in Quasiconformal Mappings.* Wadsworth and Brooks, Monterey, CA, 2nd edition, 1987.

[Bre65] J. E. Bresenham. Algorithm for computer control of digital plotter. *IBM Systems Journal*, 4:25–30, 1965.

[Bre77] J. E. Bresenham. A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, 20:100–106, 1977.

[BS95] Lev M. Bugayevskiy and John P. Snyder. *Map Projections: A Reference Manual.* Taylor and Francis, London, 1995.

[Che92] Yang Cheng. On conformal projection maintaining a desired curve on the ellipsoid without distortion. In *ACSM/ASPRS/RT '92 Technical Papers*, volume 3, pages 294–303, Bethesda, MD, April 1992.

[Che96] Yang Cheng. The conformal space projection. *Cartography and Geographic Information Systems*, 23(1):37–50, January 1996.

[Cur43] David R. Curtiss. *Analytic Functions of a Complex Variable.* Mathematical Association of America, LaSalle, Illinois, 3rd edition, 1943.

[Far93] Gerald Farin. *Differential Geometry and its Applications.* Academic Press, San Diego, 3rd edition, 1993.

[Las89] Piotr H. Laskowski. The traditional and modern look at Tissot's Indicatrix. *Cartography and Geographic Information Systems*, 16(2):123–133, April 1989.

[Pet95] Michael P. Peterson. *Interactive and Animated Cartography.* Prentice Hall, New Jersey, 1995.

[Sny94] John P. Snyder. *Map Projections: A Working Manual.* USGS/US Government Printing Office, Washington, DC, 3rd edition, 1994.

[ST83] Ian Stewart and David Tall. *Complex Analysis.* Cambridge University Press, London, 1983.

[Tob74] Waldo Tobler. Cartogram programs. Technical report, University of Michigan, Department of Geography, 1974.

[Tob86] Waldo Tobler. Pseudo-cartograms. *American Cartographer*, 13(1):43–50, 1986.

GLOBAL SCALE DATA MODEL COMPARISON

A. Jon Kimerling, Kevin Sahr, and Denis White
Department of Geosciences
Oregon State University
Corvallis, OR 97331, USA
(We acknowledge support from U.S. EPA -OSU Coop. Agreement CR 821672)

ABSTRACT

Transforming raw observations into globally regular sampling grids or surface tessellations is a fundamental data processing and storage problem underlying much of our global data analysis. The basic geometry of traditionally employed quadrilateral-based point or area grids, while well suited to array storage and matrix manipulation, may inherently hinder numerical and geostatistical modeling efforts. Several scientists have noted the superior performance of triangular point grids and associated hexagonal surface tessellations, although no thorough evaluation of global data model alternatives has been conducted. In this paper we present results from a global grid comparison study that focused on recursive tiling of polyhedral faces projected onto the globe. A set of evaluation criteria for global gridding methods were developed. Of these, metrics for spheroidal surface area, compactness, and centerpoint spacing were found to be of particular importance. We present examples of these metrics applied to compare different recursive map projection-based and quadrilateral spherical subdivision tilings. One map projection approach, the Icosahedral Snyder Equal Area (ISEA) recursive tiling, shows particular promise due to its production of equal area hexagonal tiles on the spheroid at all levels of recursive partitioning.

INTRODUCTION

A new era of high spatial and temporal resolution environmental data covering the entire globe is about to begin, ushered in by NASA's Earth Observation System (EOS) and other global data collection efforts such as the 1km AVHRR, land cover, and DEM data sets being compiled as part of the International Geosphere - Biosphere Program's Data and Information System (Eidenshink and Faundeen 1994, Hastings 1996). We should expect that earth scientists will accelerate their use of geographic information systems (GIS), numerical modeling approaches, and geostatistical methods, singly or in concert, to study global scale phenomena such as climate change and biodiversity loss. Such analyses will require both spatial and temporal integration of currently disparate data sets from a wide variety of data producers.

Transforming raw observations into global data models comprised of geometrically regular sampling grids or surface tessellations is a fundamental data processing and storage problem underlying global data analysis. One fundamental problem is that "regular" sampling grids or surface tessellations devised for the earth's surface, such as the ETOPO5 5 minute DEM or the NASA Earth Radiation Budget Experiment (ERBE) 2.5° global modeling grid, cannot be extended to the entire earth without losing regularity in both surface area and shape. Alternative approaches beg investigation.

An ancient realization is that subdividing a sphere with total regularity of surface area and polygonal shape within the tiles formed by the subdivision can be achieved only by projecting the faces of one of the five Platonic polyhedra (tetrahedron, hexahedron, octahedron, dodecahedron, icosahedron) onto the sphere. Further partitioning of any face will produce unavoidable variations in surface area, shape, or both.

Equally important is the realization that the basic geometry of commonly employed quadrilateral point grids or surface tessellations, while well suited to array storage and matrix manipulation, may inherently hinder numerical and geostatistical modeling efforts. Scientists have noted the superior performance of triangular point grids and associated hexagonal surface tessellations for numerical analyses central to studies of fluid dynamics, percolation theory, and self-avoiding walks. Additionally, hexagonal tessellations are favored by influential statisticians involved with developing survey sample designs and geostatistical methods such as Kriging .

It is clear that a thorough evaluation of alternative global data models is needed. We take a first step in this direction by presenting examples of results from a global grid comparison study funded by the U.S. Environmental Protection Agency (White et al. 1992). Comparisons are predicated on evaluation criteria, such as those presented below.

GLOBAL DATA MODEL COMPARISON CRITERIA

We believe that an ideal general purpose global data model would consist of n points and n areal cells on the globe and have the following properties:

1. Areal cells constitute a complete tiling of the globe, exhaustively covering the globe without overlapping.
2. Areal cells have equal areas.

3. Areal cells have the same topology.
4. Areal cells are the same shape.
5. Areal cells are compact.
6. Edges of cells are straight in some projection.
7. The edge between any two adjacent cells is a perpendicular bisector of the great circle arc connecting the centers of those two cells.
8. The points and areal cells of the various resolution grids which constitute the grid system form a hierarchy which displays a high degree of regularity.
9. A single areal cell contains only one point, i.e., each point lies in a different areal cell.
10. Points are maximally central within areal cells.
11. Points are equidistant from their neighbors.
12. Grid points and areal cells display regularities and other properties which allow them to addressed in an efficient manner.
13. The grid system has a simple relationship to the traditional latitude-longitude graticule.
14. The grid system contains grids of arbitrary resolution.

An early version of these criteria was formulated by Michael Goodchild, and we refer to this list as the "Goodchild Criteria". We have already noted that it is mathematically impossible for any discrete global point grid or surface tessellation to completely fulfill all of these criteria, since several are mutually exclusive. A good general purpose grid or tessellation might be expected to strike a balance among all criteria, whereas those tuned for specific applications or numerical methods might value certain of these criteria more highly. For example, geostatistical methods favor equal area tessellations that completely cover the globe and are compact.

Recursive Partitioning

Cells that can be partitioned recursively may form a tessellation system that is hierarchical and contains component sub-cells that may or may not exhibit a high degree of regularity. The terminology of recursive partitioning can best be understood from an illustration such as Figure 1. Two types of partitioning, sometimes called 4-fold and 9-fold, are shown in the top and bottom equilateral triangular cells. Each full triangle is at recursion level 0, and the initial partitioning into either 4 or 9 triangular sub-cells is termed recursion level 1. This 4-fold or 9-fold increase in the density of triangular sub-cells continues at recursion levels 2 and higher. Sets of six triangular sub-cells can be assembled into hexagons at each level of recursion, as shown in the right hand half of Figure 1. Notice that

the hexagons as assembled are symmetrical about the three triangle vertices only with 9-fold partitioning, and such symmetry is a further advantage when assembling uniform global data models. This leads us to use 9-fold partitioning in our analyses. A similar illustration could be created for recursive partitioning of spherically rectangular quadrilaterals, with the desirable symmetry present for any n-fold partitioning.

Naturally, an infinite number of triangular, hexagonal, or quadrilateral recursion levels are possible, as recursive partitioning can continue indefinitely. However, we find 9-fold partitioning to recursion levels less than ten to be suitable for comparison purposes, since surface area, compactness, centerpoint spacing, and other metrics are still computable at the rapidly increasing cell densities. However, the computation effort quickly becomes immense at higher levels of recursion and the results may not add significantly to our understanding of the surface tesselation or point grid geometry.

Evaluation Criteria Metrics

Global data model evaluation criteria are of limited practical value until metrics are developed for each. Examining the criteria presented above, we see that both topological and geometrical metrics must be devised. We have focused on geometrical measures of surface area, compactness and centerpoint spacing for both triangular and hexagonal cells on a spheroid such as the GRS80 or WGS84, although we only present results for hexagonal sub-cells in this paper. All of these measures involve determination of geodesic distances using standard ellipsoidal distance equations. Spheroidal surface area for quadrilateral cells can be computed using standard equations found in Maling (1992) and other references. Computing the spheroidal surface area of more geometrically complex cells such as spheroidal hexagons requires the oriented triangle summation approach developed by Kimerling (1984).

Measurement of spheroidal compactness proved the greatest challenge. Many two-dimensional compactness measures are based on an area to perimeter ratio normalized to 1.0 for a circle. We have extended this idea to the spheroid by determining the spheroidal area to parallel of latitude perimeter ratio, normalized to a spheroidal cap of the same surface area as the cell.

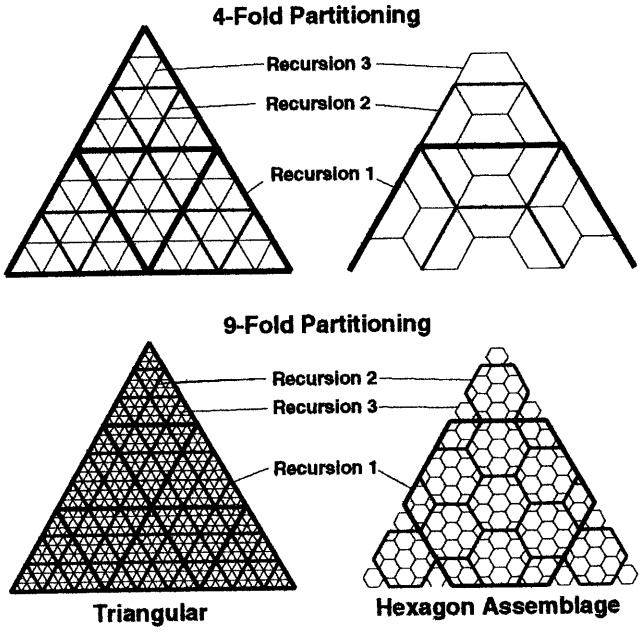


Figure 1. 4- and 9-fold triangular and hexagonal partitioning of an icosahedron face at the first three levels of recursion.

Our numerical analysis of sub-cell surface area, compactness and centerpoint distance focused on obtaining the average, range, and standard deviation for the population of sub-cells at each level of recursion. Range and standard deviation values were normalized as proportions of the average to allow direct comparison of values at different levels of recursion.

GLOBAL DATA MODEL COMPARISON EXAMPLES

Many spherical and fewer spheroidal point grids and surface tessellations have been devised as global scale data models, and in this paper we only compare a few commonly employed and/or potentially attractive tessellations. Our examples include two major classes of surface tessellations, namely quadrilateral approaches and polyhedral approaches based on map projection surfaces.

Equal Angle Quadrilateral Tessellations

Tessellations of the globe into quadrilateral cells of equal latitudinal and longitudinal extent are termed equal angle. Examples abound, including the 5' x 5' ETOPO5 global DEM, and the ERBE 2.5° x 2.5°, 5° x 5° and

10 °x 10 ° quadrilateral grids (Brooks 1981). Our example will be an initial partitioning of the globe into thirty-two 45° x 45° cells at recursion level 0, and subsequent 9-fold partitioning to recursion level 5. Hence, recursion level 2 corresponds to the 5° x 5° ERBE grid.

"Constant Area" Quadrilateral Tessellations

Constant area tessellations begin with an arbitrary sized quadrilateral cell at the equator, and then define the parallel and meridian cell boundaries across the globe so as to achieve approximately equal area cells. This is done either by keeping the latitude increment constant and adjusting the longitude increment as the pole is approached, or vice versa (Brooks, 1981). Our example is the Nimbus Earth Radiation Budget (ERB) Experiment grid, with initial 4.5° x 4.5° quadrilateral cells at the equator. The longitudinal increment increases in twelve discrete steps to 120° near each pole. Recursive subdivision into "constant area" sub-cells is more problematic, since 4-fold equal angle partitioning is commonly employed for simplicity. Breaking from tradition, we employ 9-fold equal angle partitioning to maintain consistency in our comparisons while using the same basic partitioning method. The initial 4.5° x 4.5° cells correspond approximately to recursion level 2 in the previous equal angle tessellation, and we carry the partitioning to recursion level 5.

Polyhedral Map Projection Surface Tessellations

The faces of a Platonic polyhedron are a natural starting point for a global data model, since each face is identical in surface area and is a regular spherical polygon when projected to the globe. Attention has been given to the octahedron (Dutton 1988, White et al. 1996) and the icosahedron (Baumgardner and Frederickson 1985), and we examine the latter in this paper. A convenient partitioning method for polyhedral faces is to create a map projection of each face that is of the same geometric form as the face, e.g., an equilateral triangle for each face of the icosahedron. We then partition the map projection surface recursively, producing 9 identical equilateral sub-triangles with 9-fold partitioning of the face. Sets of six sub-triangles can then be combined into hexagonal cells that are finally projected back onto the globe. Several map projections can be used, but we will examine two: the Snyder and Fuller-Gray.

The Snyder Polyhedral Equal Area projection (Snyder 1992) transforms each icosahedron face on the globe into an equilateral planar triangle while maintaining area equivalence throughout. The projection is made equal area by adjusting the scale outward from the center of each edge.

This results in increased shape distortion as each of three lines from the triangle center to corner vertices is approached.

The Fuller-Gray projection is based on the geometrical idea behind R. Buckminster Fuller's icosahedral world map projection. Fuller imagined the three edges of each icosahedron face as flexible bands curved to lie on the spherical surface. Each edge would be subdivided and holes drilled at n equally spaced increments, and flexible bands would be strung between corresponding holes on adjacent edges. This would create a triangular network of lines on the sphere, which could be flattened to create a regular grid of equilateral sub-triangles. Fuller imagined the vertices of each sub-triangle being the projection of the corresponding line intersection point on the sphere. These intersection points were later found physically impossible to achieve, since nearly all triplets of intersecting lines on the globe form small triangles in the plane, whose centerpoints are the best approximation of Fuller's idea. Gray (1994) has developed exact transformation equations for this approximation, producing a compromise projection having both small area and shape distortion. As with the Snyder projection, sub-triangles can be assembled into a hexagonal tesselation on the projection surface and globe.

GLOBAL DATA MODEL COMPARISON RESULTS

Variation in cell surface area is a major concern to geostatisticians and others. In Figure 2 we show area variation for the hexagonal and quadrilateral sub-cells produced by the four data models examined, using logarithmic scales of normalized cell areas at increasing levels of recursion with corresponding decreases in average cell area. Recognizing that the triangular partitioning of the icosahedron face performed here always creates 12 pentagons on the globe, we see that the surface area standard deviation for the Snyder projection model is always slightly greater than zero, even though there is no area variation among the hexagons forming the partition and all twelve pentagons are exactly 5/6th the area of each hexagon. However, at higher levels of recursion the 12 pentagons occupy progressively less of the total surface area and the standard deviation for the entire globe rapidly approaches zero. Hence , Figure 2 shows the Snyder model to be clearly superior for sub-cells less than 100,000 sq. km.

The variation among sub-cell centerpoint distances for the four models, seen in Figure 3, shows similar performance except for the poorly performing Equal Angle Quadrilateral model. The Fuller and Constant Area Quadrilateral models converge at higher recursion levels to essentially identical low variation, closely followed by the Snyder model.

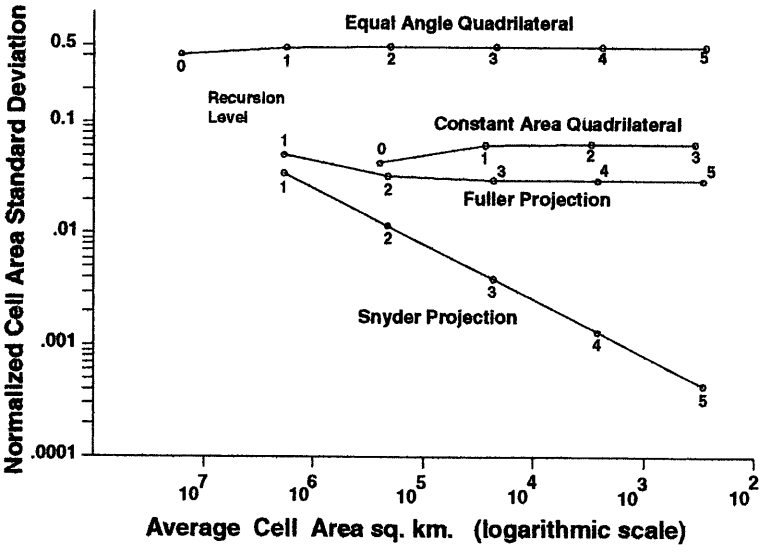


Figure 2. Normalized sub-cell area standard deviation vs. average cell area for four data models.

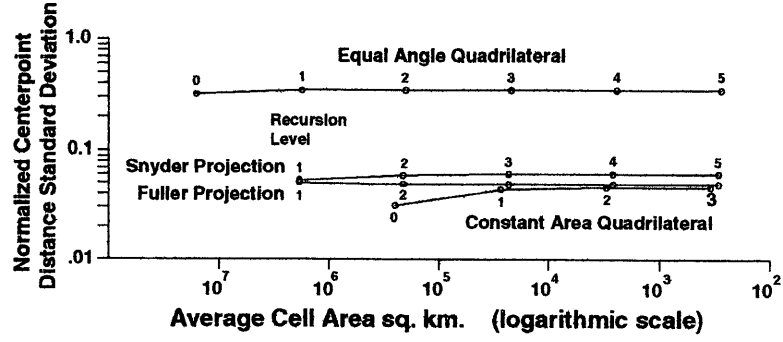


Figure 3. Normalized centerpoint distance standard deviation vs. average sub-cell area for four data models.

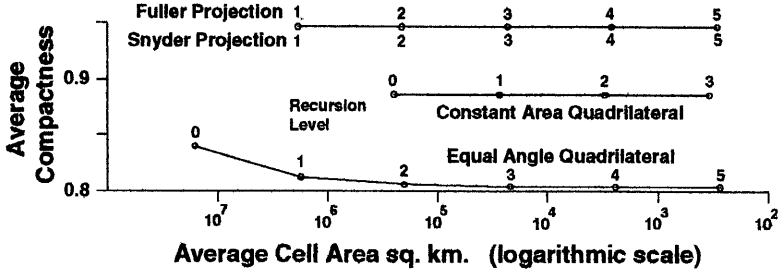


Figure 4. Average sub-cell compactness vs. average sub-cell area for four data models.

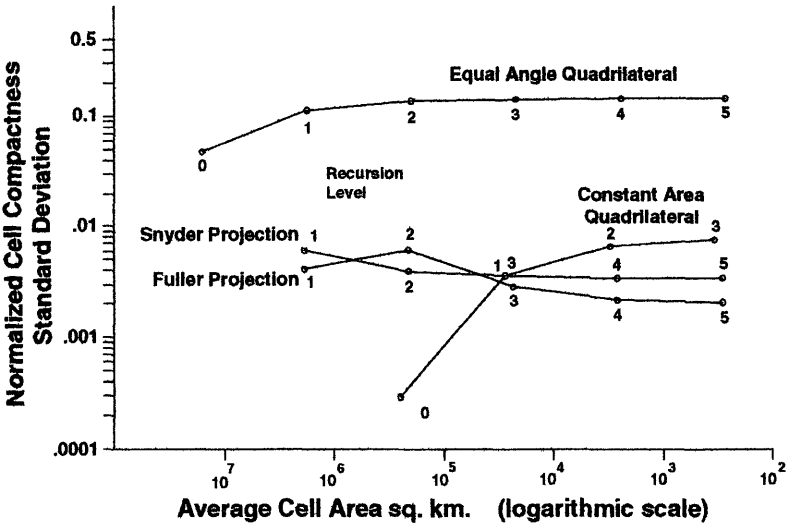


Figure 5. Normalized sub-cell compactness standard deviation vs. average sub-cell area for four data models.

Average sub-cell compactness values for the four data models (Figure 4) shows the superior performance of icosahedral models over quadrilateral, which is to be expected since hexagonal shapes are inherently more compact than rectangular. At all levels of recursion the Fuller data model produces hexagons slightly more compact than Snyder model hexagons, both being far more compact than the Constant Area and Equal Angle model quadrilaterals.

Variation in sub-cell compactness (Figure 5) shows the slightly better performance of the Fuller model over the Snyder at higher levels of recursion. The Constant Area Quadrilateral model produces the lowest variation at its initial tesselation, but compactness variation increases rapidly as the initial cells are partitioned in the equal angle manner.

CONCLUSION

Our global data model evaluation criteria and associated metrics have allowed us to compare data models varying widely in cell geometry and topology. Many more models than the four presented here as examples have been analyzed, and we have concluded that the Icosahedral Snyder Equal Area (ISEA) model recursive partitioning shows particular promise. This is due to its equal area hexagonal tiles on the spheroid, and to its high average cell compactness and low compactness variation relative to traditional quadrilateral tilings, especially at higher levels of recursion.

We are now working to develop an efficient ISEA tile addressing scheme and to demonstrate the advantages of this global data model when it is populated by data transformed from existing global data sets such as the ETOPO5 digital elevation model.

REFERENCES

Baumgardner, J.R. and P.O. Frederickson. (1985). Icosahedral Discretization of the Two-Sphere. *S.I.A.M. J. Num. Anal.* 22(6): 1107-15.

Brooks, D.R. (1981). Grid Systems for Earth Radiation Budget Experiment Applications. *NASA Technical Memorandum 83233.* 40 pp.

Dutton, G. (1988). Geodesic Modeling of Planetary Relief. *Cartographica.* 21:188-207.

Eidenshink, J.C. and J.L. Faundeen. (1994). The 1 km AVHRR Global Land Data Set: First Stages of Implementation. *Int. J. Rem. Sens.* 15:3443-62.

Gray, R.W. (1994). Exact Transformation Equations for Fuller's World Map. *Cartography and Geographic Information Systems.* 21(4): 243-46.

Hastings, D.A. (1996). The Global Land 1-km Base Elevation Digital Elevation Model: A Progress Report. *Global Change Newsletter.* No. 27: 11-12.

Kimerling, A.J. (1984). Area Computation from Geodetic Coordinates on the Spheroid. *Surveying and Mapping.* 44(4): 343-51.

Maling, D.H. (1992). *Coordinate Systems and Map Projections.* Oxford: Pergamon Press, 2nd ed.

Snyder, J.P. (1992). An Equal-Area Map Projection for Polyhedral Globes. *Cartographica.* 29(1):10-21.

White, D., Kimerling, A.J. and W.S. Overton. (1992). Cartographic and Geometric Components of a Global Sampling Design for Environmental Monitoring. *Cartography and Geographic Information Systems.* 19:5-22.

White, D., Kimerling, A.J., Sahr, K., and L. Song. (1996). Comparing Area and Shape Distortion on Polyhedral-based Recursive Partitions of the Sphere. Submitted to *Int. J. Geo. Info. Sys.*

DIGITAL MAP GENERALIZATION USING A HIERARCHICAL COORDINATE SYSTEM

Geoffrey Dutton
Department of Geography
University of Zürich
Winterthurerstrasse 190
8057 Zürich Switzerland

dutton@geo.unizh.ch

Abstract

The use of hierarchical coordinate systems in geographic information systems (GIS) is a relatively unexplored area, particularly with respect to cartographic generalization techniques. This paper describes a hybrid geospatial data model that enriches vector- topological descriptions of map features by quadtree encoding of vertex locations. It also summarizes methods to encode, analyze, filter and decode vector map data for display at scales smaller than those at which they were captured. Geometric and combinatorial computations are performed either on absolute quadtree addresses, on a world projection or directly on the sphere. The software platform presently only processes one feature class at a time, but is intended to handle more, whether stored as overlaid coverages or as independent or linked objects. Map generalization computations are localized using hierarchical hexagonal and triangular cells called Attractors. This "space-primary" approach to map generalization does not depend upon a hierarchical feature classification scheme, but the two perspectives are related and could be united. This paper describes (1) the quaternary triangular mesh (QTM) hierarchical location encoding scheme; (2) modeling of cartographic features; (3) some new generalization algorithms and conflict detection techniques; and (4) potential benefits of applying this approach across feature classes.

Hierarchical Map Generalization

Thematic Hierarchies. Approaches to hierarchical map generalization fall in two main categories. As used by some researchers (Molenaar 1996, Richardson 1994), the term refers to techniques for merging or eliminating map objects based on hierarchical feature classification. Constraints to minimum size and adjacency are normally used to eliminate or merge map features represented as polygons. The examples most often given tend to involve generalizing land use maps, which assign nominal codes to polygons at several levels of specificity, such as rural - agricultural - cropland - cornfield, or urban - industrial - transportation - railyard. Merging adjacent areas having the same use code (or removing inclusions smaller than a certain size) results in a simplified map, although the amount of line detail of the remaining polygons is not decreased accordingly. This paper does not address such possibilities, but thematic object hierarchies could potentially be used to compute semantic priority constraints for negotiating conflicts among multiple map features.

Geometric Hierarchies. Most geometric approaches to hierarchical generalization work by progressively eliminating vertices describing the importance of vertices along polylines and polygons in a consistent manner (Cromley, 1991). Doing this insures that vertices selected to define features at larger tolerances (i.e., lower resolution or bandwidth) are retained when tolerance is reduced, as additional vertices are selected which are likewise retained at yet-smaller tolerances. This also means that once a vertex is removed for display purposes, it will not reappear at smaller scales. Non-hierarchical generalization methods do not inherently include previously-selected points (nor do they always exclude previously-eliminated ones) when tolerance is changed, and this can sometimes lead to inconsistent representations, especially when zooming in and out interactively. van Osteroom (1993) and van Osteroom and Schenkelaars (1995) describe a hierarchical implementation of the widely-used Douglas line simplification algorithm (Douglas and Peucker 1973) that constructs a tree of vertices that can be repeatedly accessed to retrieve line detail at specific scales, achieving a *multi-resolution representation* (versus *multiple representations*). We take a different approach to hierarchical map generalization, using hierarchical coordinates and on-the-fly vertex selection. As in hierarchical data structures such as strip trees (Ballard 1981), various levels of detail are encoded, but being quadtree-based, it is a hierarchical partitioning of space rather than of phenomena. The space is a polyhedral approximation of a planet, and the phenomena are represented as strings of quadtree leaf addresses, each representing a two-dimensional geographic coordinate pair. This strategy combines the scale-sensitivity and indexing capabilities of quadtrees with the flexibility and rigor of vector-topological data models, as well as being able to handle data encoded as isolated features.

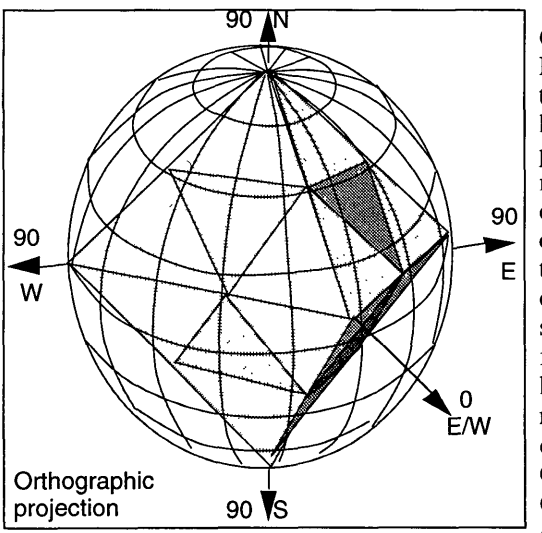


Fig. 1: An Octahedron Embedded in the Earth

Quaternary Triangular Mesh (QTM) is a global spatial indexing scheme and a hierarchical coordinate system proposed by Dutton (1989) for managing GIS positional data quality. A similar model was developed around the same time by Fekete (1990) for indexing and browsing remote sensing imagery. Various uses for QTM and related encodings have been explored by various researchers (global spatial indexing and visualization: Goodchild and Shirin 1992, Otoo and Zhu 1993; Terrain data compression: Lugo and Clarke 1995; positional data quality: Dutton 1992, 1996). A hierarchical approach to map generalization using QTM was proposed by Dutton and Buttenfield (1993), but not implemented until recently (Dutton 1996a). Work reported here further explores this line of investigation.

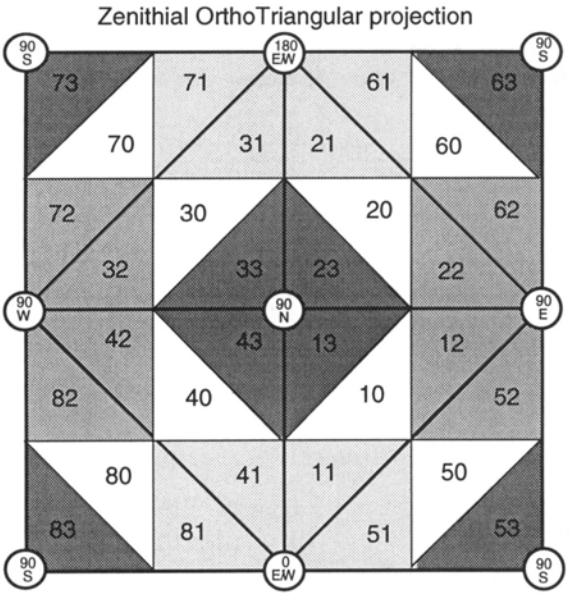


Fig. 2: QTM Octant and level 1 facet numbering

QTM Hierarchical Coordinates. To carve a planet's surface into a quaternary triangular mesh, a virtual octahedron is embedded in the earth, its vertices aligned to cardinal points (see fig. 1). The octa faces form the roots of a forest of triangular quadtrees; these eight triangles recursively bulge into four children. Child *facets* are identified by integer codes having up to 30 quaternary (base 4) digits, enabling locations on the planet as small as 2 cm^2 to be uniquely identified (spatially indexed).

The addressing scheme, while planetary in scope, is capable of handling regions smaller than land parcels, but processing time increases as precision goes up. A binary representation for QTM identifiers has been proposed (Dutton 1996) that uses 64 bits per address, usually with a number of bits left over that may be used to store other properties of points beside location. Geographic points in latitude and longitude are converted to QTM codes via an octahedral projection: more accurate points get longer addresses. The algorithm used to encode and decode point data can be implemented either recursively or iteratively. Figure 2 shows the octahedral projection and the QTM numbering scheme.

Map Encoding. Because different GIS vendors and their applications organize geographic phenomena using different schemata, QTM data processors should make as few assumptions as possible regarding data models. Therefore, all the processing techniques described below handle data at the "feature primitive" level: strings of geographic coordinates, with or without explicit topological relations. Vector map data are modeled as *features* via the *primitive elements* (point sets, polylines, polygons) that comprise them; each primitive is a set (or list) of coordinates, and features consist of lists of primitives. As a given primitive may participate in more than one feature (such as a river that serves as a property boundary), each primitive identifies the features that use it. A master catalog identifies and summarizes the locations (bounding rectangles and enclosing QTM facets) and the logical relationships of all elements. Any collection of features may be designated as a *feature set*, which can model logical, topological, positional or thematic dependencies. This storage and access architecture is diagrammed in figure 3. Note the derivation of QTM addresses and Attractor addresses from the primitives' coordinates (assumed to be latitude/longitude).

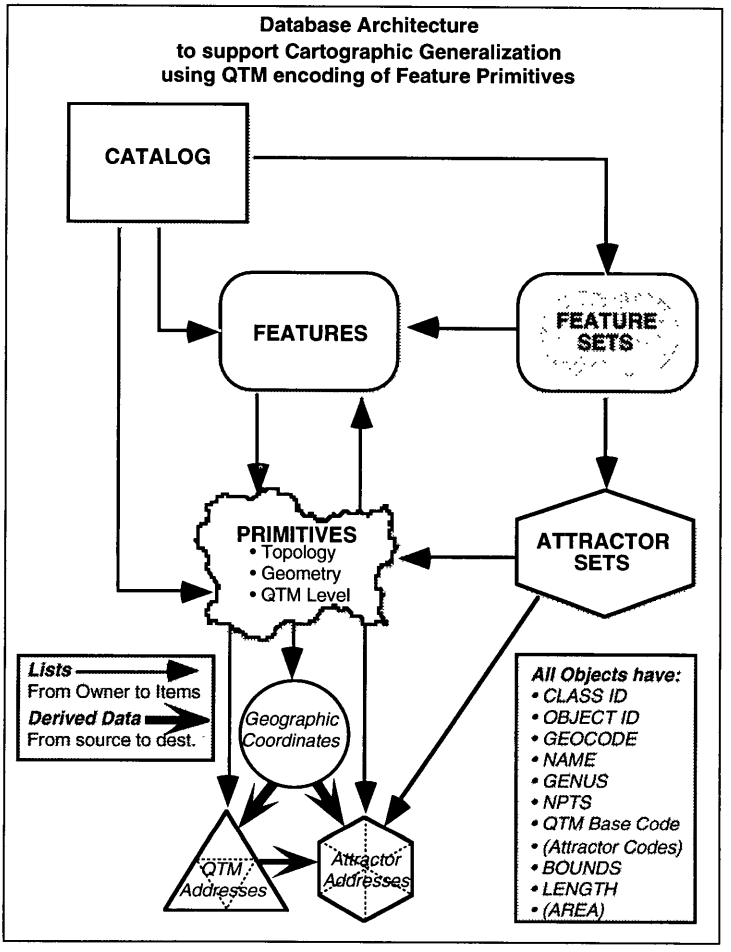


Fig. 3: Basic QTM vector feature data model

Generalizing Spatial Primitives

In preparing digital maps for generalization, all coordinates are converted to hierarchical QTM addresses, at a level of precision appropriate to their positional accuracy. This pre-processing is diagrammed in figure 4. Should positional data quality be unknown, it may be estimated by statistical analysis of QTM-filtered coordinates (Dutton and Buttenfield 1993); for medium-scale maps QTM addresses are from 15 to 25 digits long). Should features or regions be of differing accuracy, this variability can be modeled throughout processing by varying the length of QTM addresses. QTM facets group themselves into hexagonal regions, which figures 3 and 4 indicate as *attractors*. These serve as "buckets" to collect vertices and identify primitives that are likely to conflict at specific map scales. Attractors are hierarchical, but in a more complicated way than QTM facets are. They are implemented as transient, dynamic data structures (objects or lists) for conflict detection purposes only. Attractors facilitate spatial search because they contain sets of facets that are *cousins* (rather than *siblings*).

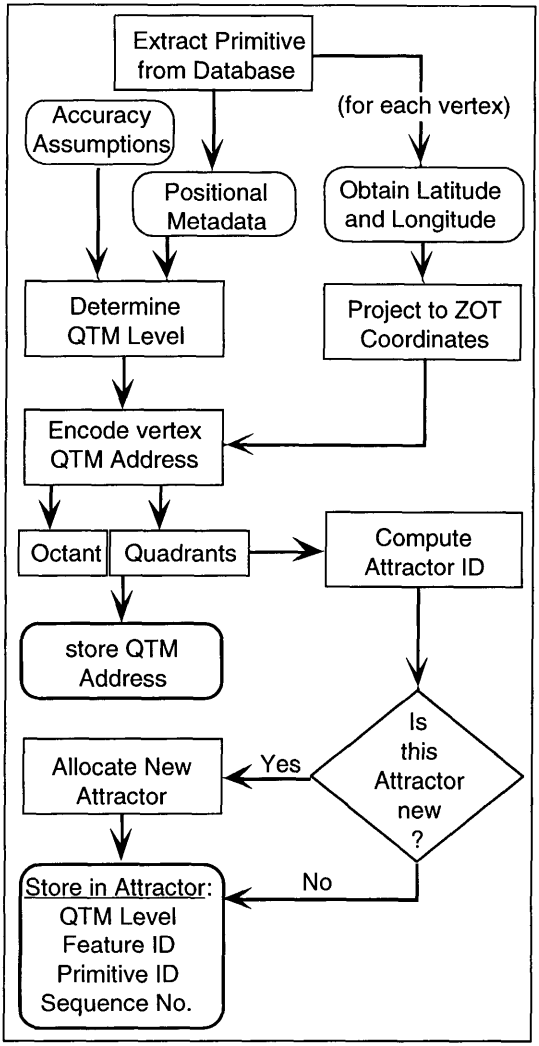


Fig. 4: Computing QTM Hierarchical Coordinates for vertices of a primitive

Whatever its source may be, the positional accuracy of map primitives needs to be expressed in linear units of ground resolution (e.g., cm) in order for QTM encoding to occur. Each QTM level of detail has a characteristic (but slightly variable) linear resolution — the edge lengths of facets making up the level. QTM encoding halts when mean resolution would drop below the linear accuracy for a primitive. To give some examples, the resolution of QTM level 17 data (76 m) is comparable to that of Landsat scenes. Level 20 resolution is about the size of a SPOT pixels (10 m), level 24 (60 cm) can resolve objects big as doormats, and level 30 (2 cm) can locate fence posts.

When digital map data is believed to be oversampled, the encoding process shown in figure 4 may include an extra step: the QTM IDs of successive vertices are compared, and duplicates are weeded out. This simple operation, when applied to attractors, is the core of a set of QTM generalization techniques.

Figure 5 shows the general structure of attractors with four levels superimposed. The triangular areas between the hexagons are also attractors; these contain single QTM facets rather than sets of six, and assist in relating the three hexagonal ones that surround them (which touch only at vertices). An actual set of attractors computed for a polygonal feature (part of the Swiss canton of Schaffhausen) is shown in figure 6 in equi-rectangular projection. Their skewed appearance reflects the shape of QTM facets in that part of the world (47.5° N, 8.5° E; attractors form perfect hexagons only at the eight QTM octant centers). Identifiers for attractors are arbitrary; currently one of the QTM facets within each attractor is selected to name it. Hence both QTM IDs and AIDs have addresses of the form OQQQ...QQQ, where O is octal and Q represents quaternary digits.

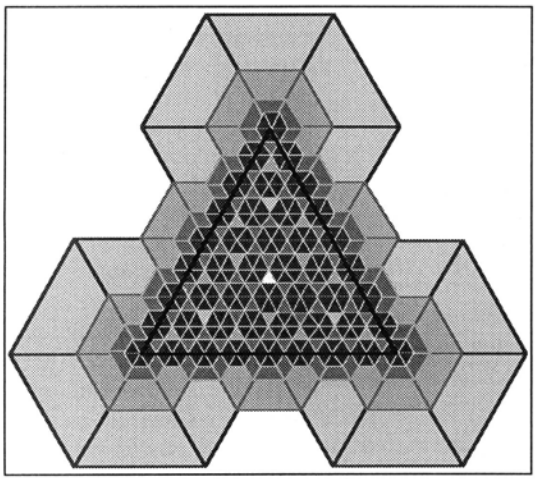


Fig 5: Four levels of attractors for a QTM facet

QTM Detail Filtering. To remove line detail when reducing a feature's scale, an appropriate level of attractor is computed for every vertex along the polyline(s) that form it, as fig. 6 shows. The basic filtering operation for individual primitives consists of scanning the sequence of vertices to determine which adjacent ones share a given attractor (non-adjacent points can also be compared). One vertex is then selected to represent all the vertices that fall in each attractor. This can be handled in various ways.

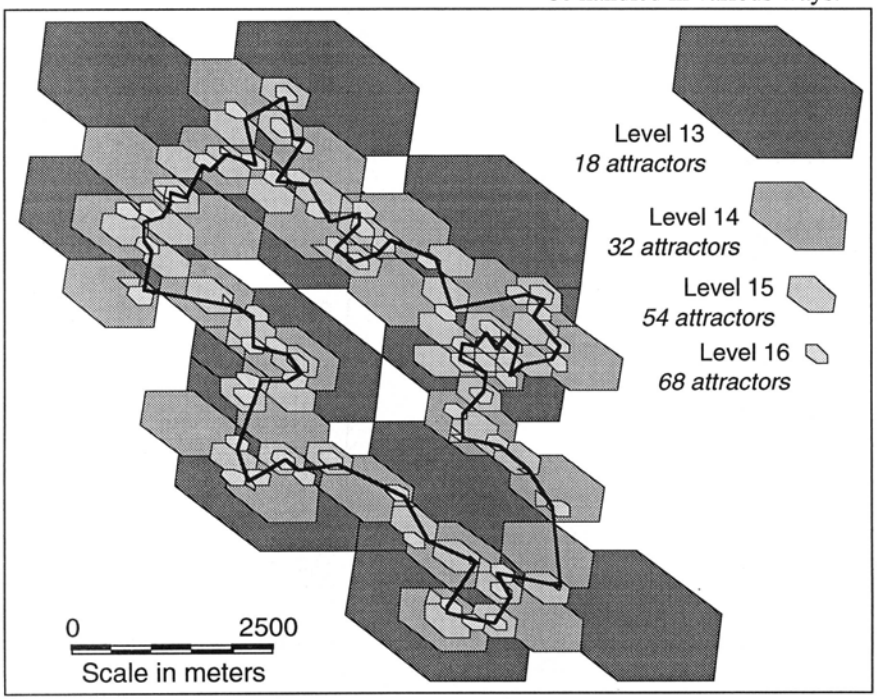


Fig. 6: Attractors occupied by vertices of a polygon having 73 points

Internally, each primitive has two string arrays allocated for it, one holding the QTM IDs of vertices, the other holding their Attractor IDs (AIDs). The array addresses are passed to a filtering function, which scans the AIDs for runs of successive duplicates. When a run is detected, a single vertex is selected from among that set of vertices to represent all of them. Which vertex is selected can make a difference in the appearance of the generalized primitive; a number of different criteria may be applied in making this choice:

1. Primitive endpoints (topological nodes) are always retained
2. Longer (more precise) QTM IDs, if any, are preferred over shorter ones
3. Vertex-specific positional metadata, if any, can be consulted
4. Longer line segments may be preferred over shorter ones
5. Sharper vertex angles may be preferred over less acute ones
6. Vertices nearest the middle of their runs may be preferred

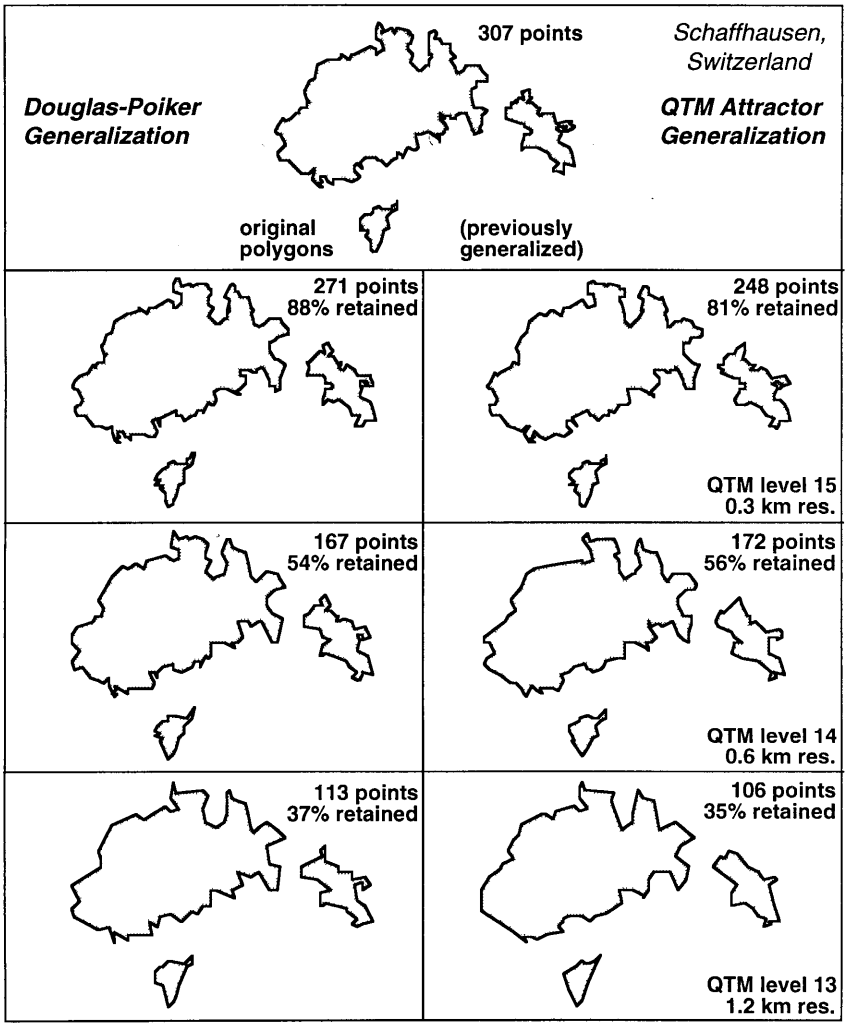


Fig. 7: Comparison of output of QTM "Median Attractor" and Douglas-Peucker Line Generalization Methods

Geometric criteria, such as items 3 and 4 in the above list, cannot be directly evaluated by functions that handle arrays of ID strings. Such evidence must be gathered when compiling QTM IDs; the results of geometric analyses can be coded into each ID in the form of *qualifiers*, as described by Dutton (1996). When metadata is unavailable, the default decision is to select the median vertex

from a run, or if there are a pair of them, the one that has the lexigraphically largest QTM address. This is somewhat arbitrary, but yields repeatable results.

Preliminary Results. The "median attractor" method of vertex elimination just described has been tested on individual map features with quite reasonable results. Figure 7 summarizes a multi-scale generalization of a Swiss canton, comparing these results to the Douglas-Peucker algorithm. The boundary data was originally digitized by the Swiss federal mapping agency in official national grid coordinates. Four files having differing levels of detail were derived from that source data using the Douglas algorithm, but the tolerance parameters were not documented. All data files were subsequently de-projected to geographic coordinates; the most detailed of these was used as input data for the QTM generalizations illustrated in fig. 7, simply by choosing a QTM hierarchical level at which to output filtered coordinates. Figure 8 displays the same results, but scaled to dimensions at which the feature might appear in printed maps.

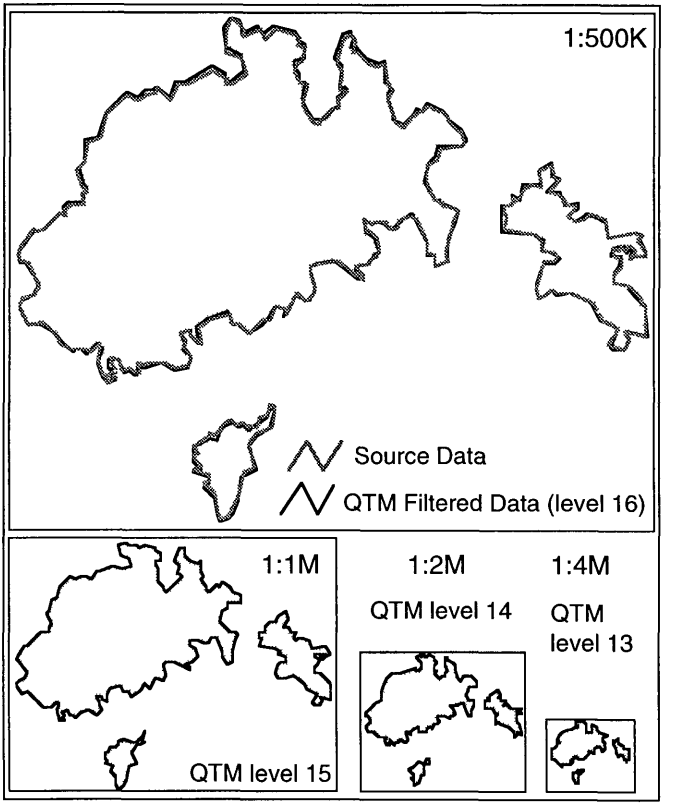


Fig. 8: QTM Generalization Results at Appropriate Scales

Multi-feature Generalization Strategies

The tests described above have dealt only with simple features, and did not explore cases where crowded features compete for map space. Multi-feature (sometimes called "holistic") map generalization is, of course, a much more difficult

problem, one which seems to require a diverse mix of strategies, as recent literature evidences (Müller et al 1995; Ruas and Plazenet 1996; Ware and Jones 1996). Most prior work in this area uses one or more "object-primary" techniques, which the nature of vector-topological GIS data structures makes necessary; proximity relations between spatial objects must be explicitly (and expensively) modeled to detect scale-related conflicts. The alternative is to use "space-primary" methods, which are normally restricted to raster data environments; in them, space is modeled and objects are attributes of locations. Few approaches to vector-based, space-primary map generalization have been developed (see Li and Openshaw 1993 for a rare example), but this perspective may have considerable heuristic value. Populating lists of QTM attractors with pointers to primitives that occupy them is one way to combine space- and object-primary approaches to detect conflict among any number of features. Choosing the size (level) of attractors allows one to cast as fine or coarse a net as one wishes, and QTM's spatial indexing properties can be used to restrict the search space.

It is already possible to export scale-specific, QTM-filtered versions of geodata (i.e., multiple representations) to GIS databases. Eventually, QTM-encoded map data may reside in GIS databases themselves, providing built-in multi-resolution capabilities. To make either approach work, additional processing and decision-making will be necessary to generalize QTM-encoded features for display, as different selections of features (and different purposes and applications) will require continually revisiting regions of interest to make new decisions about how to portray them. How to resolve cartographic conflicts may never be easy to decide, but at least we will know what they are and where to find them.

References

Ballard DH 1981. Strip Trees: A hierarchical representation of curves. CACM 24(5): 310-21

Cromley RG 1991. Hierarchical methods of line simplification. Cartography and Geographic Information Systems 18(2): 125-31

Douglas DH, Peucker TK 1973 Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10(2): 112-22

Dutton G 1989. Modelling locational uncertainty via hierarchical tessellation. Accuracy of Spatial Databases, M. Goodchild & S. Gopal (eds.). London: Taylor & Francis, 125-140

Dutton G 1992. Handling positional uncertainty in spatial databases. Proc. Spatial Data Handling Symp. 5. Charleston, SC, August 1992, v. 2, 460-469

Dutton G, Buttenfield BP 1993. Scale change via hierarchical coarsening: Cartographic properties of Quaternary Triangular Meshes. Proc. 16th Int. Cartographic Conference. Köln, Germany, May 1993, 847-862

Dutton G 1996. Improving locational specificity of map data: A multi-resolution, metadata-driven approach and notation. Int. J. of GIS. London: Taylor & Francis, 253-268

Dutton G 1996a. Encoding and handling geospatial data with hierarchical triangular meshes. Proc. 7th Symp. on Spatial Data Handling. Delft NL, August 1996, 8B.15-28

Fekete G 1990. Rendering and managing spherical data with sphere quadtrees. Proc. Visualization '90 (First IEEE Conference on Visualization, San Francisco, CA, October 23-26, 1990). Los Alamitos CA: IEEE Computer Society Press

Goodchild MF, Yang Shiren 1992. A hierarchical data structure for global geographic information systems. CVGIP, 54:1, pp. 31-44

Li Z, Openshaw S 1993. A natural principle for the objective generalization of digital maps. CaGIS 20(1): 19-29

Lugo JA, Clarke KC 1995. Implementation of triangulated quadtree sequencing for a global relief data structure. Proc. Auto Carto 12. ACSM/ASPRS, 147-156

Molenaar M 1996. The role of topologic and hierarchical spatial object models in database generalization. In Molenaar M (ed) Methods for the generalization of geo-databases. Publications on Geodesy, Delft, Netherlands Geodetic Commission 43: 13-36

Müller J-C, Lagrange J-P, Weibel R 1995 (eds) GIS and generalization: methodological and practical issues. London, Taylor & Francis

Otoo EJ, Zhu H 1993. Indexing on spherical Surfaces using semi-quadcodes. Advances in Spatial Databases (Proc. 3rd Int. Symp. SSD'93), Singapore, 510-529

Richardson D E 1994. Generalization of spatial and thematic data using inheritance and classification and aggregation hierarchies. In Waugh, TC, Healey RG (eds) Advances in GIS research (Proceedings Sixth International Symposium on Spatial Data Handling): 901-20

Ruas A, Plazanet C 1996 Strategies for automated generalization. In Kraak M J, Molenaar M (eds) Advances in GIS research II (Proceedings 7th Int. Symp. on Spatial Data Handling): pp. 6.1-6.18.

van Oosterom P 1993. Reactive data structures for geographic information systems. Oxford, Oxford University Press

van Oosterom P, Schenkelaars V 1995. The development of a multi-scale GIS. Int. J. of GIS 9(5): 489-508

Ware J M, Jones C B 1996 A spatial model for detecting (and resolving) conflict caused by scale reduction. In Kraak M J, Molenaar M (eds) Advances in GIS research II (Proc. 7th Int.Symp. Spatial Data Handling): 9A.15-26

AN EVALUATION OF FRACTAL SURFACE MEASUREMENT METHODS USING ICAMS (IMAGE CHARACTERIZATION AND MODELING SYSTEM)

Nina Siu-Ngan Lam
Professor, Department of Geography & Anthropology
Louisiana State University
Baton Rouge, LA 70803, USA

Hong-lie Qiu
Assistant Professor, Department of Geography
California State University, Los Angeles
CA 90032-8222, USA

Dale Quattrochi
Geographer, NASA Marshall Space Flight Center
Huntsville, AL 35812, USA

ABSTRACT

With the fast pace of increase in spatial data anticipated in the EOS (Earth Observing System) era, it is necessary to develop efficient and innovative tools to handle these data. ICAMS (Image Characterization and Modeling System) is an integrated software module designed to provide specialized spatial analytical functions for visualizing and characterizing remote-sensing data. Fractal analysis is the main module in ICAMS. Although fractals have been studied extensively before, the question of which fractal measurement method should be used remains. This paper evaluates the three fractal surface measurement methods that have been implemented in ICAMS, including the isarithm, variogram, and triangular prism methods. Results from applying five simulated surfaces of known dimensions (D = 2.1, 2.3, 2.5, 2.7, and 2.9) to the three methods show that the isarithm method calculates the fractal dimensions fairly accurately for all surfaces. The variogram method, on the other hand, yields accurate results only for surfaces of low dimensions. For surfaces of higher dimensions, the variogram method is unstable. The triangular prism method produces inaccurate results for almost all the surfaces, and its usefulness is questionable. More in-depth evaluation, however, is needed to verify the present findings.

INTRODUCTION

We are currently working on the development of a software module called ICAMS (Image Characterization and Modeling Systems). ICAMS is designed to run on Intergraph-MGE and Arc/Info platforms to provide specialized spatial analytical functions for characterizing remote-sensing images. The main functions in ICAMS include fractal analysis, variogram analysis, spatial autocorrelation analysis, texture analysis, land/water and vegetated/non-vegetated boundary delineation, temperature calculation, and scale analysis.

The development of ICAMS has been driven by the need to provide scientists efficient and innovative spatial analytical tools for characterizing and visualizing large-scale spatial data such as remote-sensing imagery. As spatial data become increasingly available, the need for useful analytical tools to analyze these various forms of spatial data becomes more pressing. The NASA's Earth Observing System (EOS) to be launched in the late 1990's is one example data source that will provide useful data to the scientific community. The fast pace of increase in digital data posts an immediate problem, that is, how such an enormous amount of data can be handled and analyzed efficiently. Clearly, advances in global as well as local environmental modeling must need both components: data; and the analytical tool to handle the data. An overview of the theoretical background of and the practical need for developing ICAMS, as well as its system design and functionality, can be found in Quattrochi, et al. (1997).

Along with the need for more efficient and innovative spatial analytical techniques is the need for more fundamental research on the applicability and reliability of such techniques. Through the employment of an integrated software package such as ICAMS, it would be easier to carry out the evaluation tasks, and by making the software available to the wider scientific community, a variety of applications and evaluations can be made. These advantages will be realized especially in ICAMS, as most of the implemented specialized functions have seldom been applied to landscape characterization using remote-sensing imagery, though they were considered to have great potential in characterizing landscape patterns for global environmental studies (Woodcock and Strahler 1987).

This paper focuses on the use of the fractal module in ICAMS. In particular, we examine the three fractal surface measurement methods that have been implemented in the software, including the isarithm, variogram, and triangular prism methods. A series of hypothetical fractional Brownian motion (fBm) surfaces with known fractal dimensions were first generated. These surfaces were applied to the three algorithms in ICAMS on the Intergraph-MGE platform to compute their fractal dimensions. The comparison between the known and the computed fractal dimensions provides an assessment of the

reliability and effectiveness of the three most commonly used fractal surface measurement methods for characterizing and measuring landscape patterns.

The evaluation results will be useful to further improvement of the fractal measurement methods and possible modifications of the algorithms in ICAMS. A host of related research questions utilizing fractals can be examined. For example, do different environmental /ecological landscapes and processes (e.g. coastlines, vegetation boundaries, wetlands) have their unique fractal dimensions? Can the fractal dimension be used as a means to identify regions with different properties, and ultimately be used as a part of metadata? Or, what is the significance of changes in fractal dimension, either in time or space?

METHODS AND DATA

Fractal analysis has been suggested as a useful technique for characterizing remote sensing images as well as identifying the effects of scale changes on the properties of images (De Cola 1989 & 1993; Lam 1990; Lam and Quattrochi 1992). A major impediment in applying fractals is that there are very few algorithms readily available for researchers to use and experiment, and for those who can access or directly construct their own programs, the frustration is that the results from applying differing algorithms often contradict each other. A thorough evaluation of the various measurement techniques is necessary before they can be used to reliably characterize and compare the various types of landscapes.

The three fractal surface measurement methods that have been implemented in ICAMS, the isarithm, variogram, and triangular prism methods, have been applied to real data and documented in detail in various studies (e.g., Lam and De Cola 1993; Jaggi et al., 1993). However, they have never been systematically evaluated using controlled, synthetic surfaces. The use of controlled surfaces in testing these algorithms, such as the fractional Brownian motion (fBm) surfaces used in this study, should provide a standard to compare with and therefore helps in revealing the major characteristics and differences among the methods. For the ease of interpretation, the following provides a brief description of the three methods as implemented in ICAMS.

The isarithm method, sometimes also called the walking-divider method, utilizes the isarithms of the surface as a means in determining the fractal dimension D of the surface, where $D_{surface} = D_{isarithm} + 1$. The algorithm was evolved from Goodchild (1980), Shelberg, et al. (1982), and Lam and De Cola (1993). In addition to the data matrix with the numbers of rows and columns specified (note that the number of rows does not have to be the same as the number of columns), the isarithm method in ICAMS requires the following parameter input by the user: the number of steps or walks, the isarithmic interval, and the direction from which the operation proceeds (either row, column, or both).

For each isarithmic value and each step size, the algorithm first classifies each pixel below the isarithmic value as white and each above this value as black. It then compares each neighboring pixel along the rows or columns and examines if the pairs are both black or both white. If they are of different colors, then there is an isarithm lying between the two neighboring pixels. The length of each isarithm line is approximated by the total number of boundary pixels. It is possible for a given step size that there are no boundary pixels. In this case, the isarithm line is excluded in the calculation. The total number of boundary pixels for each step size is plotted against step size in log-log form, also called the fractal plot, and a linear regression is performed. The regression slope b is used to determine the fractal dimension of the isarithm line, where $D = 2 - b$. The final D of the surface is the average of the D values for those isarithms for which $R^2 \geq 0.9$. Figure 1 shows a typical output from the isarithm method in ICAMS on the Intergraph-MGE platform.

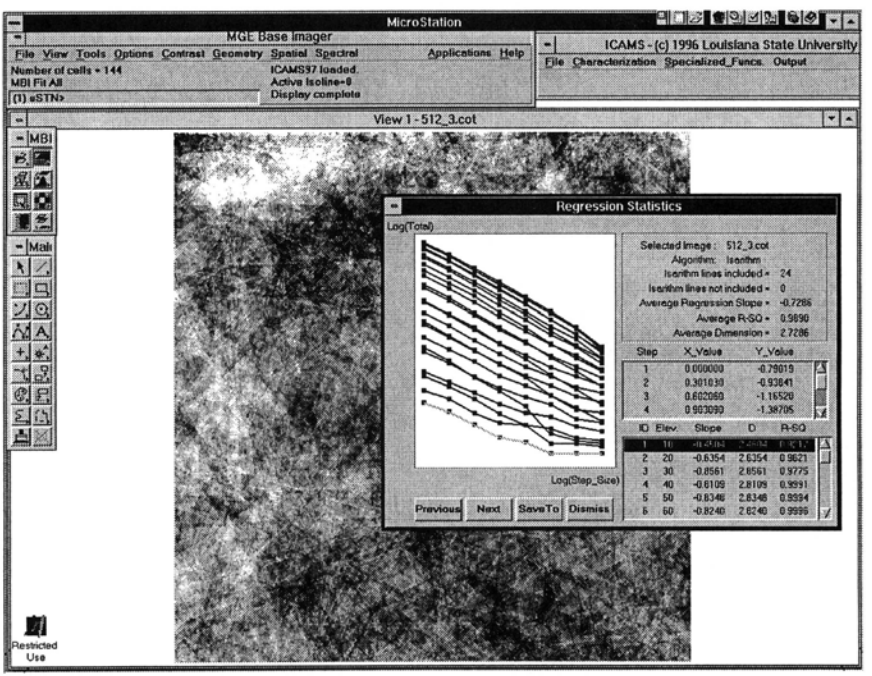


Figure 1. An output from the isarithm method. The background image is a simulated surface with $D = 2.7$ (see discussion below).

In the variogram method, the variogram function, which describes how variance in surface height varies with distance, is used for estimating the fractal dimension. The only difference between the traditional variogram and the variogram used in fractal estimation is that distance and variance are portrayed in double-log form. The slope of the linear regression performed between these two variables is then used to determine the fractal dimension, where in this case, $D = 3 - (b/2)$. Mark and Aronson (1984) pioneered the use of the variogram method. Detailed discussion of the method can also be found in Lam and De Cola (1993) and Jaggi, et al. (1993). In ICAMS, the variogram method requires the following parameter input: the number of distance groups for computing the variance, the sampling interval for determining the number of points used in the calculation, and the sampling method (regular or stratified random). Sampling only a subset of points for calculation is necessary especially for large data sets such as remote-sensing imagery, as the computational intensity will increase dramatically with increasing number of data points. Figure 2 shows an output from the variogram method.

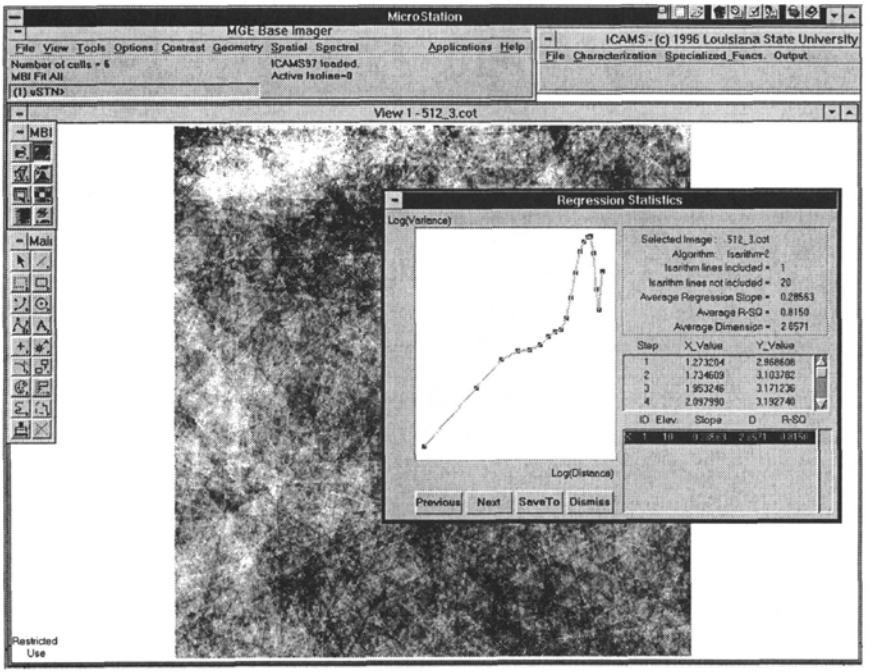


Figure 2. An output from the variogram method using the same simulated surface as Figure 1.

The triangular prism method compares the surface areas of the triangular prisms with the pixel area (step size squared) in log-log form (Clarke 1986; Jaggi et al. 1993). For each step size, the triangular prisms are constructed by connecting the heights of the four corners of the pixel to its center, with the center height being the average of its four corners. The areas of these surfaces can be calculated by using trigonometric formulae. The fractal dimension is calculated by performing a regression on the surface areas and pixel areas, where $D = 2 - b$. Figure 3 is an example output from the triangular prism method.

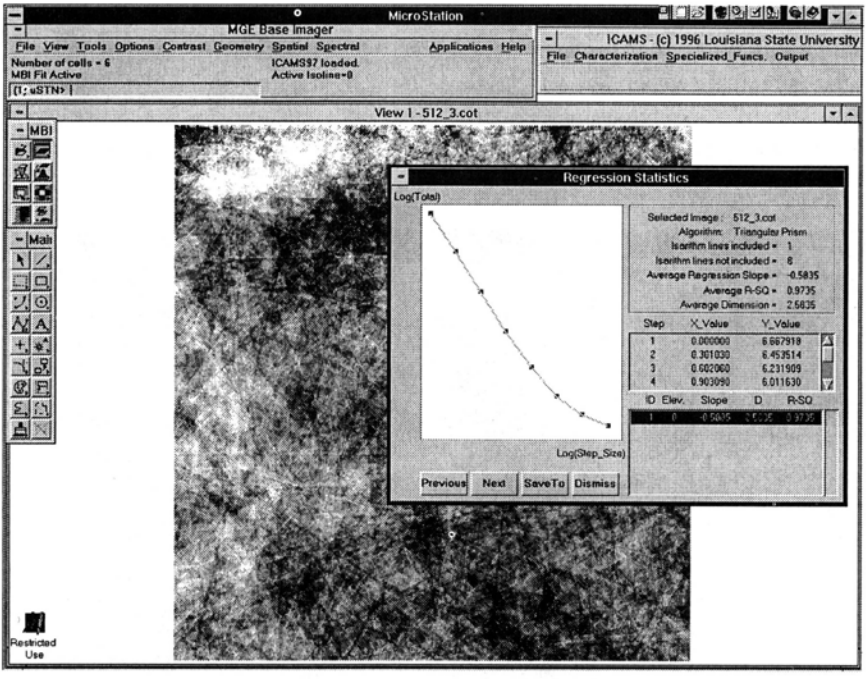


Figure 3. An output from the triangular prism method using the same simulated surface ($D = 2.7$).

To test the three fractal surface measurement methods, we use the shear displacement method to generate a series of hypothetical surfaces with varying degrees of complexity (i.e., fractal dimension) (Goodchild 1980; Lam and De Cola 1993). The method starts with a surface of zero altitude represented by a matrix of square grids. A succession of random lines across the surface is generated, and the surface is displaced vertically along each random line to form a cliff. The process is repeated until several cliffs are created between adjacent sample points. The amount of displacement is controlled by the variogram parameter H in such a way that the variance between two points is proportional

to their distance scaled by H. H describes the persistence of the surface and has values between 0 and 1, and the fractal dimensions of the simulated surfaces can be determined by $D = 3 - H$. The value $H = 0.5$ ($D = 2.5$) results in a Brownian surface.

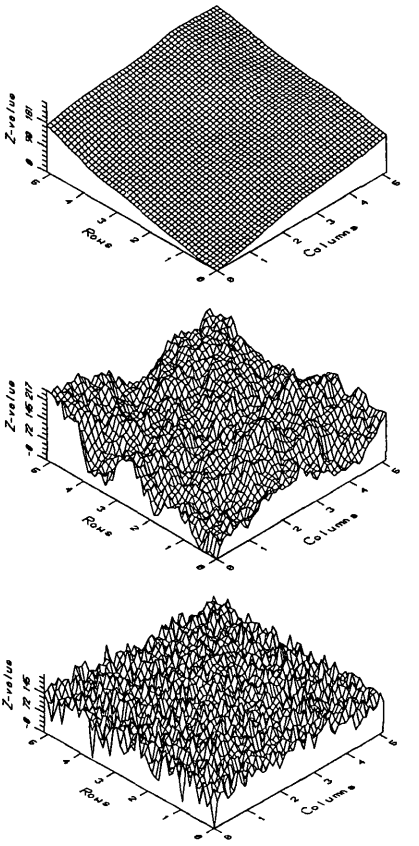


Figure 4. Three simulated surfaces from top to bottom, $D = 2.1$, 2.5, and 2.9.

Five surfaces with $H = 0.1$, 0.3, 0.5, 0.7, and 0.9 were generated for this study. Each surface has 512 x 512 rows and columns and was generated with 3000 cuts and the same seed value for the random number generator. Figure 4 displays three of the simulated surfaces ($H = 0.1$, 0.5, 0.9 or $D = 2.9$, 2.5, 2.1). The Z values of all simulated surfaces have been normalized so that their minimum and maximum Z values are 0 and 255. These surfaces are input to ICAMS for fractal calculation. For the isarithm method, the parameter input are 8 step sizes with an isarithmic interval of 10. In the variogram method, number of distance groups were fixed at 20, with a sampling interval of 10 using the

stratified random sampling method. The only parameter in triangular prism is the number of steps, which was also fixed at 8.

RESULTS AND DISCUSSION

The results from applying the five hypothetical surfaces to ICAMS are summarized in Table 1. Table 1 shows that as D increases (or H decreases), the standard deviations of the surface values decrease. The inverse relationship between D and standard deviation is notable, because D is considered as a measure of spatial complexity and standard deviation a measure of non-spatial variation.

Table 1: Summary of results for the five simulated surfaces. R^2 values are in parentheses. The isarithmic algorithm includes Row, Col. and Both and all have $R^2 > 0.90$ and therefore R^2 are not listed.

D	H	Mean	SD	Row	Col.	Both	Variogram	Tiangular
2.9	0.1	110	26	2.93	2.99	2.96	2.85 (0.57)	2.73 (0.97)
2.7	0.3	113	30	2.73	2.90	2.79	2.88 (0.63)	2.58 (0.97)
2.5	0.5	118	48	2.53	2.57	2.54	2.59 (0.84)	2.31 (0.98)
2.3	0.7	112	67	2.27	2.13	2.21	2.21 (0.99)	2.10 (0.98)
2.1	0.9	121	75	2.14	Nil	2.05	2.09 (0.99)	2.10 (0.86)

The isarithm method in ICAMS generally performs very well for all five surfaces, with the computed fractal dimension agreeing with the dimension values used in simulating these surfaces. There are some discrepancies in resultant dimension values when using different orientations (row, column, and both). Such difference may be attributed to individual surface characteristics, where some surfaces may have more features with distinct orientations, such as roads, canels, or agricultural fields. In fact, the availability of an orientation option in this method could help in disclosing these individual surface characteristics that are otherwise not obvious.

The variogram method yields accurate results for surfaces of low fractal dimensions, but its performance becomes unstable with increasing dimensionality. Perfect fit ($R^2 = 0.99$) occurs in surfaces of $D = 2.1$ and 2.3, which are also the dimensionality of most real-world topographic surfaces. For surfaces of higher dimensions, the variograms do not behave linearly in the log-log plot. The user would have to determine through eye-balling only a range of points that looks reasonably linear to be included in the regression. For example, for the surface of $D = 2.7$ (Figure 2), if only the first 9 points are included in the regression, then D becomes 2.92 with a $R^2 = 0.94$, which is

different from the result in Table 1 when all 20 points are included in the regression (D = 2.88, R^2 = 0.63).

The performance of the triangular prism method is disappointing, with the computed dimensions being consistently lower than the known dimension. Similar findings have also been reported in Jaggi, et al. (1993).

Based on the results from this analysis, we may conclude that the variogram method may not be a good measurement method for most remote-sensing imagery, as they tend to yield much higher dimensions than topographic surfaces. The variogram method, however, would be a useful method for computing fractal surfaces of low dimensions. Our findings on the reliable performance of the isarithm method, however, are contrary to those of Klinkenberg and Goodchild (1992), where the divider methods were reported to have extremely disappointing performance due to their inability to discriminate visibly different surfaces. More studies are needed to verify the initial findings.

CONCLUSION

The three fractal surface measurements methods implemented in ICAMS, including the isarithm, variogram, and triangular prism methods, were evaluated using five simulated surfaces of varying degrees of complexity. The results show that the isarithm method yields accurate and reliable results for all surfaces, whereas the variogram method is only accurate for surfaces of low dimensions such as topographic surfaces. The use of variogram method for remote-sensing imagery is questionable, as the images are generally of much higher dimensions than topographic surfaces. The triangular prism method is the most inaccurate as it does not yield similar fractal dimension values. We will in the near future perform more evaluation to confirm the results from this study.

ACKNOWLEDGEMENT

This research is supported by a research grant from NASA (Award number: NAGW-4221). We thank graduate assistants Rajabushananum Cherukuri and Wei Zhao for their technical assistance.

REFERENCES CITED

Clarke, K. C. (1986). Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method. *Computers & Geosciences* 12(5):713-722.

De Cola, L. (1989). Fractal analysis of a classified Landsat Scene. *Photogrammetric Engineering and Remote Sensing* 55(5):601-610.

De Cola, L. (1993). Multifractals in image processing and process imaging. In *Fractals in Geography*, N. S.-N. Lam and L. De Cola, eds. Prentice Hall: Englewood Cliffs, NJ, pp. 282-304.

Goodchild, M. F. (1980). Fractals and the accuracy of geographical measures. *Mathematical Geology* 12:85-98.

Jaggi, S., Quattrochi, D. A., and Lam, N. S.-N. (1993). Implementation and operation of three fractal measurement algorithms for analysis of remote-sensing data. *Computers & Geosciences* 19(6):745-767.

Klinkenberg, B. and Goodchild, M. F. (1992). The fractal properties of topography: a comparison of methods. *Earth Surface Processes and Landforms* 17:217-234.

Lam, N. S.-N. (1990). Description and Measurement of Landsat TM images using fractals. *Photogrammetric Engineering and Remote Sensing* 56(2):187-195.

Lam, N. S.-N. and Quattrochi, D. A. (1992). On the issues of scale, resolution, and fractal analysis in the mapping sciences. *The Professional Geographer* 44(1):89-99.

Lam, N. S.-N. and De Cola, L., eds. (1993). *Fractals in Geography*. Prentice Hall: Englewood Cliffs, NJ., 308p.

Mark, D. M. and Aronson, P. B. (1984). Scale-dependent fractal dimensions of topographic surfaces: an empirical investigation, with applications in geomorphology and computer mapping. *Mathematical Geology* 11:671-684.

Quattrochi, D. A., Lam, N. S.-N., Qiu, H. L., and Zhao, W. (1997). Image characterization and modeling systems (ICAMS): a geographic information system for the characterization and modeling of multiscale remote-sensing data. In *Scale in Remote Sensing and GIS*, D. Quattrochi and M. Goodchild, eds. Boca Raton, FL: CRC/Lewis Publishers, pp. 295-307.

Shelberg, M. C., Lam, N. S.-N., and Moellering, H. (1983). Measuring the fractal dimensions of surfaces. *Proceedings, Sixth International Symposium on Automated Cartography (Auto-Carto 6)*, Ottawa, Canada, Vol. 2, pp. 319-328.

Woodcock, C. E. and Strahler, A. H. (1987). The factor of scale in remote sensing. *Remote Sensing of Environment* 21:311-332.

SIMULATING AND DISPLAYING SURFACE NETWORKS

Falko T. Poiker, BASc
School of Engineering Science
fpoiker@sfu.ca

Thomas K. Poiker, Professor
Department of Geography
poiker@sfu.ca

Simon Fraser University
Burnaby, B.C.
Canada, V5A 1S6

ABSTRACT

This paper deals with surface structures using valleys (channels) and ridges of fluvially eroded terrain. The basic proposition is that they are topological duals. Both channel and ridge networks are binary trees, with the ocean the root of channel trees and high mountaintops the root of ridge trees.Given the topological relationship between the two networks, it is straight-forward to estimate the approximate geometric position of one network from the other.

To estimate one network from the other, we have to have a notion of the surface between the two networks which we will call "slope behavior" here. The fall-line or slope-line will be used here as representative for the slope behavior. A crude approximation to the fall-line could be a straight line. But that is usually not enough. A more general approximation would be an s-curve or concavo-convex profile. Of course, in some cases, components of the s-curve could degenerate to zero.

The approach has many applications: 1. It can serve as a testbed for physical geographers to visualize different assumption about erosion and slope shape. 2. It can speed up digitizing efforts by encoding less than the full terrain and estimating the missing components. 3. It can provide more natural terrain for animation sequences.

INTRODUCTION

This paper is a combination of two efforts:

- Representing terrain
- Using a novel structure of terrain

The attempt to give relief the impression of depth is an old cartographic undertaking. Wiechel (1878) developed the first method for relief shading. The idea was used extensively by cartographic artists but the analytical method had to

wait until computers made the many calculations feasible (Yoeli, 1965; Horn, 1982; Peucker and Cochrane, 1974). Their approximations through linework (Tanaka 1930; Tanaka 1950; Peucker, Tichenor, and Rase 1974) were successful while line plotters dominated graphic production but have not been used for the last twenty years. The belief that different colors appear at different distances to the viewer let Karl Peucker (Peucker, 1908) develop a system of altitudinal tints that were supposed to give a plastic impression.

All these were orthographic representations of terrain which in cartography were considered superior because of the ability to measure horizontal position and distance. But with the rapid evolution of computer graphics and its emphasis on camera-like graphics, cartography went along with this field's push for an increasing use of perspective views. For the last decade, this issue has been in the hands of the computer graphics community.

What is left for the GIS community is to develop realistic or at least believable models of terrain.

MODELLING TERRAIN

There are many ways a computer can be used to generate terrain. The challenge is to allow control over the general look of the terrain. How much control is needed depends on the wishes of the user. For GIS purposes, where the graphical representation must match real terrain as closely as possible, as much of the terrain's features as possible has to be controllable.

In computer graphics, where the user does not need to represent an existing terrain, landscapes are often generated using fractal brownian methods (Mandelbrot, 1982; Fournier et all, 1982). The entire landscape is random. While fractal methods are commonly used, they give no control to the terrain creator with respect to the locations of the mountains or valleys.

The Ridge-Channel Concept

Digital Terrain Models as a subject within GIS has seen a passage through four levels of structures. This passage shows the evolution of structures from highly machine-oriented ones to those that attempt to integrate expert knowledge of terrain.

The first level, lasting for a fairly short period, was the attempt to duplicate some manual methods of data gathering. Terrain profiles across street-lines to compute cut-and-fill values are the most frequently mentioned. (Miller and Laflamme, 1958)

The second level, that of the regular grid, was initially used because of its simple geometry and the ease with which it could be adapted to loop-oriented early programming languages, its main drawbacks being the large data volumes and the distortion of terrain that all but the finest grid created. These problems have recently been eliminated by the development of very efficient compression algorithms which have given this approach a new license.

As an alternative to the regular grid, different types of surface approximations were developed. The one that has survived the period and is still in use is the Triangulated Irregular Network (TIN) (Peucker, et al. 1976), which divides surfaces into triangular facets. The edges often follow breaks in the surface, thus eliminating one of the major weaknesses of the regular grid.

The fourth and so far final level creates meta-structures of terrain. Typically, this is by describing terrain by networks, either of contours or of valley- and ridge-lines (Mark, 1977). This can be a very powerful structure because it incorporates much of our knowledge about surfaces.

The study of rivers as networks (or rather trees) is a branch of Geomorphology that is relatively old. Arthur Strahler suggested a hierarchy of rivers (Strahler, 1956). Werner showed that besides river networks, there are also ridge networks and they are topological duals (Werner, 1977; Pfaltz, 1976). In other words, for every pair of rivers that meet at a confluent, there is a ridge that originates in such a confluent or at least nearby and between the two rivers Maxwell (1870) and later Warntz (1966) developed a different structure that did not have confluents.

To give the above statement more precision, we should elaborate. First, river and ridge systems only develop on "fluvially eroded surfaces". Even though these types of surfaces cover the largest part of the earth's land, there are areas where the statements do not hold. We will disregard these areas. It is appropriate to exchange the term "river" with the term "channel" as the more general term. A channel does not have to carry water all the time, yet it has the valley shape that is so important for this consideration. As one follows a channel from the source to the end, the run-off becomes shallower, relating the logarithm of the height to the channel's length. Both channel and ridge networks are binary trees, with the ocean the root of channel trees and mountain tops the roots of the ridge trees. Incidentally, channels and ridges are often called "surface-specific lines" (Peucker et al., 1976).

Given the topological relationship between the two networks, it is straightforward to deduce the approximate geometric position of one network from the other. In practice, the estimation of the ridge network from the channel network is the more likely one to happen since the latter is more recognizable from maps and other materials and easier to digitize.

To deduce one network from the other, we have to have a notion of the surface between the two networks which we will call "slope behavior" here. The fall-line or slope-line will be used as representative for the slope behavior. A crude approximation to the fall-line could be a straight line. But that is usually not enough. A more general approximation would be an s-curve which would be defined by three lines of certain lengths and angles (the upper, middle and lower arms) plus, if desired, some smoothness factors at the junctions. Of course, in some cases, components of the s-curve could degenerate to zero.

In the early stages of a river, material is eroded by water and ice and transported away. A V-shaped valley (only the middle arm non-zero) is the result. Toward the river mouths, the water slows and deposits some of the debris that was carried down. The river plane which is being built is the lower arm of the s-curve. When a glacier moves through a valley, it cuts into the valley, leaving the upper parts of the old valley intact and building a usually wide, u-shaped valley: all three arms are well developed. In the case of canyons, the upper arm is totally flat and so is the lower, with the middle arm being nearly vertical.

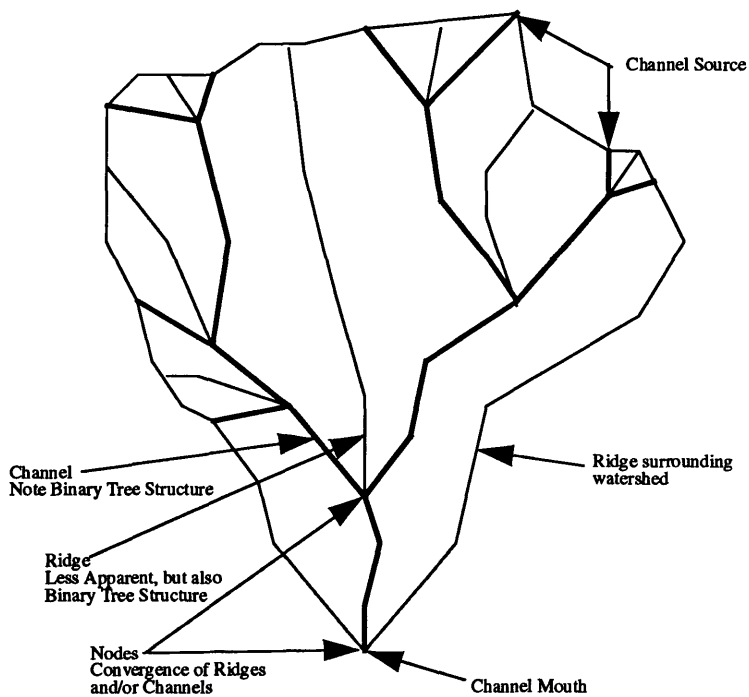


FIGURE 1. Example of a Ridge Channel Network

THE RIDGE-CHANNEL STRUCTURE

Neither the Maxwell/Warntz nor the Werner approach specified the Ridge Channel properties sufficiently for our work. we therefore had to put forward some of our own properties:

1. Ridges bisect the angle formed by two Channels emanating from a Channel confluent.
2. Channels end at Ridges, and the root of the binary tree of Channels starts at a meeting point of two Ridges.
3. If one follows a Channel from a node to its end (at a Ridge) and then follows the Ridge back to the node, one will make one loop of what we call a Face. A Face is a polygon that has as its edges Ridges on one side and Channels on the other. Encircling a Face involves one change from Ridge to Channel. A Face has no Ridges or Channels inside. This is true for any Face defined in a Ridge Channel network.
4. A node will consist of one of the following: 3 Channels and 1 Ridge; 1 Channel and 2 Ridges (at the outside of a watershed), 2 Channels and 2 Ridges (at the inside of a watershed), 0 Channels and 3 Ridges, 1 Channel and 3 Ridges.

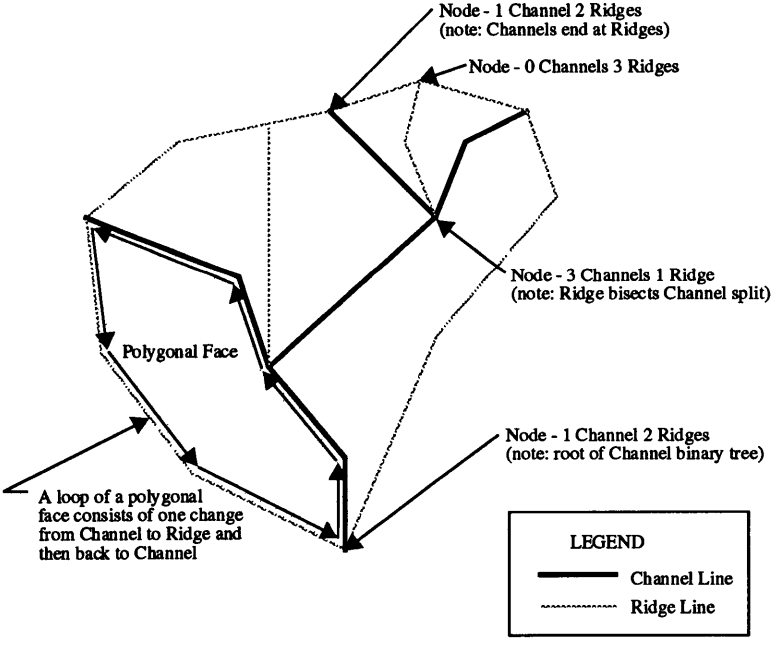


FIGURE 2. Ridge Channel Properties

The most important property mentioned above is that of the Face. By using well known polygon data structures, the algorithm can take advantage of the power of these data structures in building the surface representation. Since a Face always consists of 1 Channel (or rather, a series of Channel lines linked together) and 1 Ridge (or rather a series of Ridge lines linked together), the Face becomes a good starting point for calculating the resulting surface.

The algorithm begins by reading in the Ridge Channel data in line segments. At the same time, it builds up a winged edge data structure (Baumgart, 1972), a structure very similar to POLYVRT (Peucker and Chrisman, 1975) and other topological polygon structures. A winged edge data structure is a polygon data structure which uses pointers to link edges of one polygonal face together. The edge structure contains pointers to the two polygons it partially defines, pointers to the two end coordinates that define it, and pointers to the next and previous edges on each polygon. The pointers are the "wings" of the winged edge data structure. Using a winged edge structure ensures several things:

1. The validity or completeness of the data can be verified by checking (among other things) the nodes,
2. Ridges and Channels can easily be moved within the structure, due to the same properties of the winged edge data structure.
3. Faces can be encircled easily
4. The structure can easily be defined recursively - with greater detail, one face can become another Ridge Channel network. This ensuing network can be linked to the lesser detailed Faces by using a few extra pointers (or "wings").

One of the drawbacks of the winged edge structure is that the algorithm does not make clear to which polygon the next and previous edges belong. This was solved by using a structure which is similar to the split edge data structure which defines two records for each edge - one for one polygon (and the corresponding next and previous edges) and one for the other polygon (and the corresponding next and previous edges). The structure still has only one record, but sorts the pointers so that the first next edge, previous edge and polygon pointers correspond to each other, and the second next edge, previous edge and polygon pointers correspond to each other. The structure looks as follows:

```
typedef struct w_edge
{
    coordinate    *end[2];     // array of two coordinates define the ends of
                               // the edge.
    polygon       *poly[2];    // array of two polygon pointers - polygon[0]
                               // and polygon[1].
    w_edge        *next[2];    // next[0] corresponds to polygon[0], next[1]
                               // corresponds to polygon[1].
    w_edge        *prev[2];    // prev[0] corresponds to polygon[0], prev[1]
                               // corresponds to polygon[1].
} winged_edge;
```

Figure 3 is a diagram of the modified winged edge data structure:

Once the structure is complete, the terrain is generated on a face by face basis. For each face, the nodes which defines the transition between the Ridge of the Face and the Channel of the Face are identified. The Ridge and the Channel are divided into equal segments depending on the resolution needed for the picture and, at each subdivision interval, a strip is generated from the Channel to the Ridge using the slope behavior as a guide.

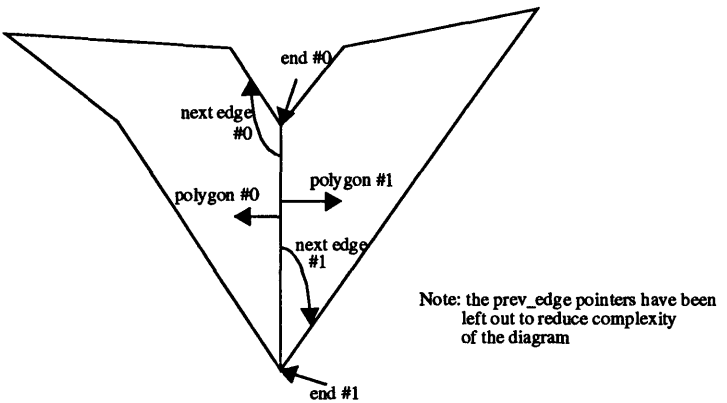


FIGURE 3. Pointer sorting in Winged Edge data structure

The new algorithm:

```
ridge_channel()
{
    /* read in from a file in the form: (x,y,z) (x,y,z)... */
    read_in_ridges_and_channels();

    /* nodes stored in a linked list with pointers to the winged edge data structure */
    store_node_locations();

    /* for each edge, find it's left and right adjacent edge at both ends and set the
       edge pointers accordingly. Node pointers are also resorted for easier traversal */
    set_winged_edge_pointers();

    /* each node is checked for validity, */
    check_node_validity();

    /* setup the polygon data structure - polygons have a pointer to one of its edges and
       are stored in a linked list of polygons */
    build_polygons();
```

```
    /* the polygons are now each traversed and drawn */
    draw_polygons();
}
```

Slope Behavior

Werner(1988) argues that Channels(C) and Ridges (R) are related through the valley side slope (V, our slope behavior), so that C + V = R and R + V = C, where "the plus sign represents the combination of two bodies of information" (p 253). The combination C + R = V is not possible because the same ridges and channels can be linked by different slopes.

Without giving justice to the finer points of slope geomorphology but following the advice of some knowledgeable geomorphologists, we postulate that all slopes between ridges and channels can be approximated by an S-curve. The S-curve is defined by three lines of varying lengths and angles (the upper, middle and lower arms) plus, if desired, some smoothness factors at the junctions. As described above, some of the arms can be reduced to zero.

At this stage of the research, we have restricted ourselves to three types of slope behavior: 1. Young erosion or the V-shaped valley, 2. The glacial or U-shaped valley, and 3. The canyon shape with the upper and lower arms being horizontal and the middle arm being vertical. These slope types are hard coded in the program. Later, a slope generation layer will be inserted, allowing the user the choice of form or deducing the behavior from digitized fall-lines.

CONCLUSION

This model obviously needs refinement. Certain results of such a simple structure don't look right. For refinement, we need the input of the geomorphologist and at present, they are not very interested in this type of landscape treatment. But such a framework has the advantage of providing visualizations that can be readily compared. It allows experimentation with different parameters. It also delivers landscape views for animation that are clearly better than the present fractal models supply.

LIST OF REFERENCES

BAUMGART, B.G. 1972. Winged-edge Polyhedron Representation. Techn. Rep. STAN-CS-320, Computer Science Department, Stanford University, Palo Alto, CA. Cited in Foley, JD, A van Dam, SK Feiner and JF Hughes, 1990. Computer Graphics, Principles and Practice. Reading, etc.

FOURNIER, A., D. FUSSELL and L. CARPENTER, 1982. Computer Rendering of Stochastic Models. CACM: 25(6), June, 371 - 384

HORN, B.K.P. 1982. Hill shading and the Reflectance Map. *Geo-Processing, 2(1), October, 65 - 144*

MANDELBROT, B. 1982. *The Fractal Geometry of Nature.* San Francisco.

MARK, D.M. 1977. *Topological Randomness of Geometric Surfaces.* Ph.D. thesis, Simon Fraser University.

MAXWELL, J C, 1870. On Hill and Dales. *Phil. Mag.*, 40: 421 - 427

MILLER C L and R A LAFLAMME, 1958. The Digital Terrain Model - Theory and Application. Photogr. Engineering, 24(3), 433 - 442

PEUCKER, K. 1908. *Beitraege zur Geschichte und Theorie der Gelaendedarstellung.* Wien. Reprinted Amsterdam 1970.

PEUCKER, T.K. and N. CHRISMAN, 1975. Cartographic Data Structures. *American Cartographer,* 2: 55 - 69

PEUCKER, T.K. and D. COCHRANE 1974. Die Automation der Reliefdarstellung - Theorie und Praxis. *Int. Yearbook of Cartography, 1: 128 - 139*

PEUCKER, T.K., M. TICHENOR, and W.-D. RASE 1974. The Computer Version of Three Relief Representations. J.C. DAVIS and M. McCULLAGH, Eds. *Display and Analysis of Spatial Data.*

PEUCKER, T.K., R.J. Fowler, J.J. Little, and D.M. Mark 1976. Triangular Irregular Networks for Representing Three-Dimensional Surfaces. *Technical Report Number 10, Geographical Data Structures Project, CNR Contract N00014-75-C-0886*

PFALTZ, J.L., 1976. Surface Networks. *Geographical Analysis,* 8, 77 - 93

STRAHLER, A.N. 1956. Quantitative slope analysis. *Bulletin of the Geological Society of America* 67: 571-606.

TANAKA, K. 1930. A New Method of Topographical Hill Delineation. *Memoirs College of Engineering, Kyushu Imperial University,* 5(3), 121 - 143

TANAKA, K. 1950. The Relief Contour Method of Representing Topography on Maps. *Geographical Review,* 40: 444 - 456

WARNTZ, W., 1966. The Topology of a Socio-Economic Terrain and Spatial Flows. *Papers, Regional Science Assoc.*, 17: 47 - 61

WERNER, C., 1977. *Towards a General Theory of Maturely Eroded Landscapes.* Unpublished Manuscript

WERNER, C., 1988. Formal Analysis of Ridge and Channel Patterns in Maturely Eroded Terrain. Annals, AAG, 78 (2), 253 - 270

WIECHEL, H., 1878. Theorie und Darstellung der Beleuchtung von nicht gesetzmaessig gebildeten Flaechen mit Ruecksicht auf die Bergzeichnung. *Civilingenieur*, 24, 335-364

THE PROBLEM OF CONTOUR IN THE GENERATION OF DIGITAL TOPOGRAPHIC MAPS

Silvania Avelar
Remote Sensing Center
Federal University of Minas Gerais
30161-970 Belo Horizonte, Brazil
silvania@csr.ufmg.br

ABSTRACT:

Topographic maps are generated using Digital Terrain Models (DTMs), which provide the basis for numerical solutions of several important problems, such as the determination of contour lines of the terrain. Because DTMs do not address the question of the shape of the region worked with, in certain cases they may represent the region imprecisely. This work is concerned with the contour problem in the generation of topographic maps. By contour we mean a simple polygon that bounds a region containing all points gathered in the terrain. This paper presents a technique to determine a contour using geometric characteristics of the terrain data.

1 INTRODUCTION

Problems involving terrains are well documented in the literature (Van Kreveld, 1996). Work in this area, besides being useful for society, is especially interesting for computational geometry.

Several mathematical models have been used to represent the terrain numerically, but they usually do not take into account the shape of the region worked with. In general, algorithms for terrain modeling consider the convex hull (Preparata and Shamos, 1985) of the set of points In doing so, when the region of interest is not convex, they can induce wrong results. Although this question has been subject of intense research in computer science, in this scenario it remains without a suitable solution, to the best of the author's knowledge. The work presented in the following sections is a contribution in this direction.

This work address the contour problem in the generation of topographic maps. By contour we mean the polygonal curve, not necessarily convex, that bounds the polygonal region containing all points gathered in the terrain. We will focus the contour problem in the contour lines layer of topographic maps.

We begin by briefly presenting aspects related to the digital generation of topographic maps. Then, we describe a solution to the following problem: given n points $p_1, p_2, \ldots, p_n$ of the plane, compute the boundary polygon which fits the region containing these points better than the convex frontier. The aim is to minimize the imprecision in the representation of a terrain.

This paper is organized as follows. The next section discusses the Digital Terrain Models and the contour problem. Section 3 presents the methodology used to generate topographic maps considering contour determination. Section 4 describes a contour-generating algorithm. The following section makes some practical considerations about the implementation and shows one result obtained. Finally, concluding remarks are presented in the last section.

2 DIGITAL TERRAIN MODELS AND THE CONTOUR PROBLEM

For generation of digital topographic maps, a mathematical model describing the terrain is required, and this description must be as close as possible to the terrain's real aspect. In general, a Digital Terrain Model (DTM) is composed by points sampled from the region under study.

Digital Terrain Models are classified according to the mathematical model used. Interpolation models (network of points/tesselations) are usually preferred to approximation models (analytical equations) (Sakude, 1992). Based on the spatial distribution of the sampled points, the models can have a regular distribution (square, rectangular and triangular tesselations) or an irregular one. Despite their frequent use, regular tesselations do not yield a good representation of the variations of the terrain, because they are created artificially (Buys et alli, 1991). Tesselations of irregular distributions based on the original points gathered in the terrain can define more precisely the region in study. Because the triangle is the minimum polygon, irregular tesselations are usually triangulations.

There are many possible different triangulations for the same point set. Intuitively, a "good" triangulation for the propose of terrain modeling is the one in which triangles are as equiangular as possible. In other words, it is desirable to avoid long and thin triangles (De Floriani, 1987; Falcidieno and Spagnuolo, 1991; Buys et alli, 1991).

The Delaunay triangulation, a fundamental construction in Computational Geometry, is as equiangular as possible, and for this reason it is a standard tool in terrain description (Preparata and Shamos, 1985). Besides its very good capability of terrain modeling, it also saves on computation time with the choice of a suitable data structure. However, the domain of the Delaunay triangulation is the convex hull of the point set, and in certain situations the region of interest is not convex. Thus, another kind of frontier is necessary. In the case of sinuous regions like roads, for instance, this is a serious problem, because computations are extrapolated to places not known in the original region. In practice, this can invalidate the resulting topographic maps.

In this context, it is necessary to determine a contour to points set, minimizing extrapolation errors. We present a mechanism that minimizes this type of problem, while using the Delaunay triangulation. In the next sections we describe an algorithm which dynamically modifies the original convex hull to address this situation.

3 METHODOLOGY

Using the coordinates x, y and z of the points gathered in the terrain, the adopted method to the generation of topographic maps, considering their contours, consists in five fundamental steps (Figure 1).

1. Partition of the region into triangles, using the Delaunay triangulation of the sampled points.
2. Determination of a polygon smaller than the convex hull contouring the point set.
3. Elimination of the Delaunay triangles outside the new contour.
4. Computation of the points that will constitute the contour lines for the representation of the relief.
5. Design of the topographic map: B-Spline interpolation of the constituting points of the contour lines and insertion of information of other required layers.

A number of well-known algorithms exist to implement each of the steps above, except for step 2, in which we will work.

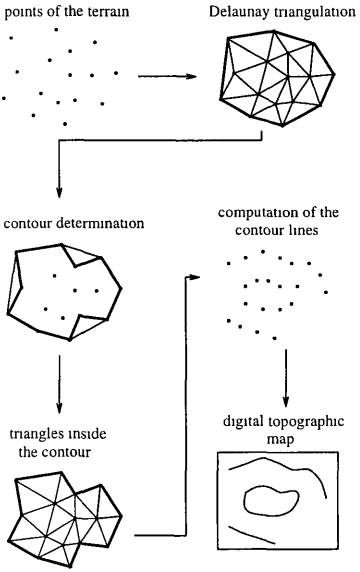


Figure 1· The applied methodology.

4 THE CONTOUR-GENERATING ALGORITHM

4.1 Definitions

For a given finite set of points of the plane, we wish to determine a simple polygon, with a smaller area than the convex hull, which bounds the polygonal region containing the points.

A polygon is defined as an ordered sequence of n ($n \geq 3$) points in the plane, $p_1, p_2, \ldots, p_n$, and the edges $p_1p_2, p_2p_3, \ldots, p_{(n-1)}p_n$ and p_np_1 formed by them A simple polygon is a polygon with the restriction that non-consecutive edges do not intersect (Shermer, 1992) The convex hull of the point set defines the convex polygonal region with the smallest area that contains the points.

4.2 Description of the algorithm

The algorithm determines the contour departing from the known convex hull of the points. The idea is to dynamically modify the convex frontier, looking for candidate points to constitute the new edges of the searched contour. A circle is used to determine the candidate points to be analysed.

Beginning at one of the edges of the convex hull, a circle having this edge as diameter is drawn. The points of the set that lie within this circle are the candidate points. A candidate edge for the new contour is obtained by joining the candidate point closest to one of the two vertices of the edge under consideration to that vertex to which it is closest. To verify whether this is an acceptable edge, we form a triangle with the two vertices that defined the circle and the candidate contour point. Should the triangle so formed not contain any other point, the chosen point is accepted and the candidate edge will form part of the final polygon sought. In this way, the edge which was being worked with can be discarded and replaced by two new ones, reducing the area delimited by the contour.

This process is repeated recursively to one of the generated edges until the circle drawn contains no other points In this case, the edge is kept and the process goes on to the next edge of the convex hull. On completion of the process, the contour sought is produced. The convex polygon may be processed in a clockwise or anti-clockwise direction. The direction in which the polygon is processed affects the shape of the final contour.

A more formal description of the contour-generating algorithm is:

1. Let $c_1, c_2, \ldots, c_n$ be the vertices of the determined convex hull, whose edges are ordered as $c_1c_2, c_2c_3, \ldots, c_nc_1$. The coordinates of the vertices $c_1, c_2, \ldots, c_n$ are $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ respectively.

2. Each of the edges of the convex hull is worked on separately. To start the algorithm, one vertex of the convex hull is chosen as c_1 (for example, that vertex with the smallest y coordinate value) and a direction (clockwise or anti-clockwise) is chosen for proceeding the algorithm Starting from the edge c_1c_2, for example in the anti-clockwise direction, the circle C is determined whose diameter is given by the length of the edge being worked with and whose center lies at the mid-point of this edge.

3. The circle C will establish a region for analysis equivalent to a half-circle in which may lie points which will determine reentrances, defining edges different from the previous one. Then it is determined which points (x, y) lie within C.

4. Considering the points inside C, it is determined the closest one, called p_t, to one of the extremities c_1 or c_2:

   (a) The candidate edge is formed by p_t and the vertex nearest to p_t.

   (b) It is determined whether points exist within the triangle formed by c_1, c_2 and p_t.

   (c) Should any point lie within $c_1c_2p_t$, the point p_t is eliminated from the analysis and the candidate edge is not accepted. The next point meeting the condition set in item 4 is then identified

   (d) Should $c_1c_2p_t$ not contain any other point in its interior, p_t will form part of the solution polygon, defining an edge with the vertex to which it is nearest, c_1 or c_2.

5. A new circle C is drawn, with diameter equal to the edge formed by p_t and the anterior vertex which it is not nearest.

6. Steps 4 and 5 are repeated until no point lies within the circle, which means that the edge that defined the circle, in the context in question, can not be further reduced.

Figure 3 illustrates the working of the contour-generating algorithm applied to the edge c_1c_2 of the convex hull of the set of points in Figure 2.

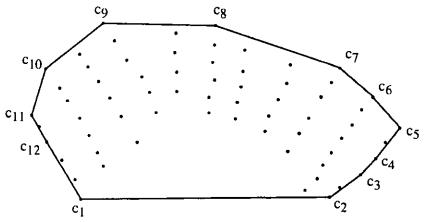


Figure 2. The convex hull of a finite set of n points.

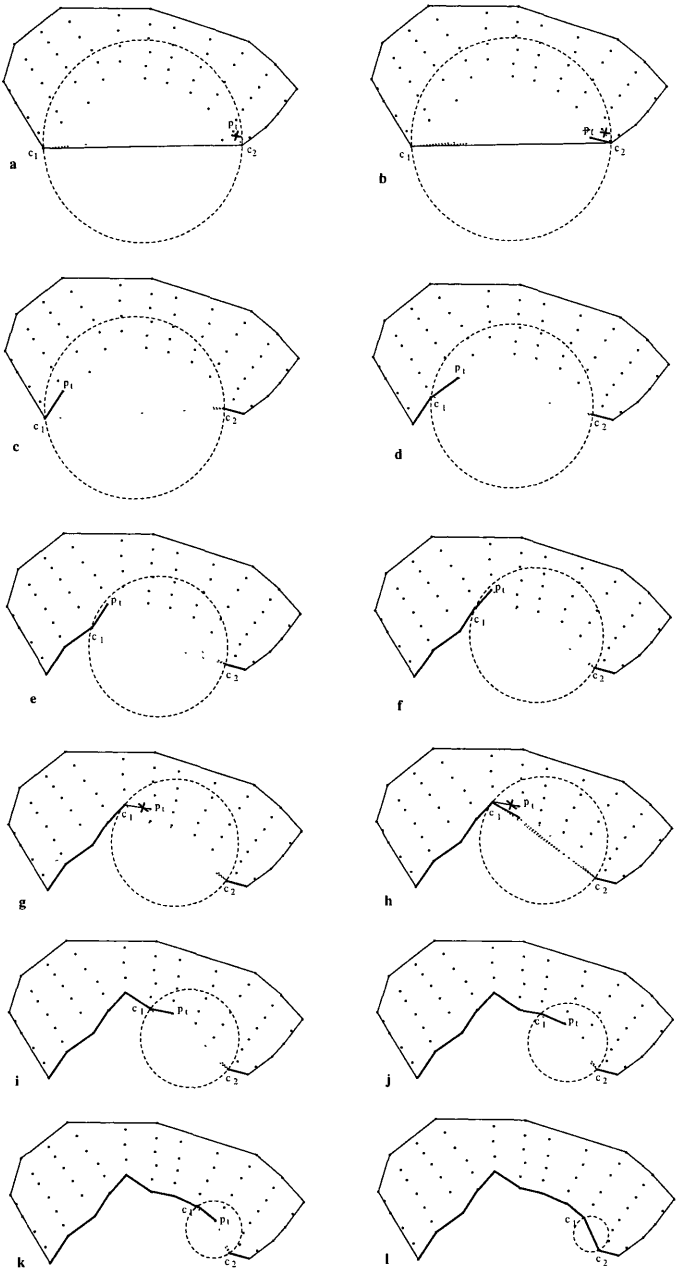


Figure 3: Contour generating for the edge c_1c_2 of the convex hull.

4 PRACTICAL CONSIDERATIONS

The described algorithm was coded as a module in a system designed to generate topographic maps in engineering projects related to roads With this, it was possible to do tests with real data and a more realistic model. It was used in the generation of topographic maps in the project of duplication of the Brazilian interstate road BR-381 between the states of Sao Paulo and Minas Gerais Figure 4 shows the contour lines generated using the described method with contour determination for a topographic map from this project

The system was implemented in C and the compiler used was the Gnu C. It can work in DOS and Unix The algorithm to implement the Delaunay triangulation was based on the divide-and-conquer approach (Lee and Schachter, 1980), using the winged-edge data structure (Baumgart, 1975)

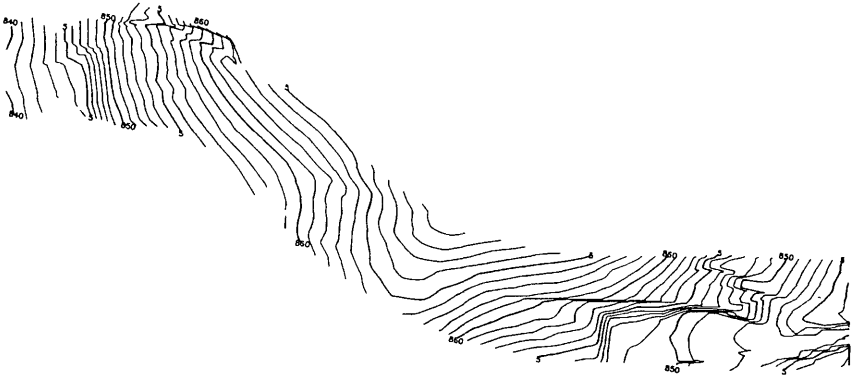


Figure 4. An example of contour determination.

5 CONCLUDING REMARKS

Many applications in Geographic Information Systems and other areas, as computer graphics and robotics, require a polygonal form closer to the region of points being dealt with, that is, they require that a non-convex contour of the points be generated In order to meet the needs of such applications and minimize imprecision in representation of the set of points, this algorithm can be used In most cases, the described method proved to be appropriate, finding a smaller polygon than the convex one. It also presented a good tradeoff between the quality of the solution and time, being able to solve complex instances on a PC compatible microcomputer (Avelar, 1994).

As the direction in which the polygon is processed affects the shape of the final contour, it can be considered that there is an alternative solution which can be compared with the first one, chosing the more convenient contour to the worked region.

This solution is not, in general, the optimal one, because of the very strong convexity hypothesis. The optimality criteria depends on each particular application Of course, if the polygonal region is convex, the convex hull is the right answer. But, in general, this is not true, as illustrated in Figure 3. The new shape has vertices and edges that do not belong necessarily to the original polygon

BIBLIOGRAPHY REFERENCES

Avelar, S (1994) *Representation of Relief using Geometric Algorithms* (In Portuguese) Master's Thesis, Federal University of Minas Gerais, Dept. of Computer Science.

Baumgart, B.G. (1975). A Polyhedron Representation for Computer Vision. *AFIPS Proceedings*, Vol.44, pp 589-596.

Buys, J., H.J. Messerschmidt and J.F. Botha (1991) Including known discontinuities directly into a triangular irregular mesh for automatic contouring purposes. *Computer & Geosciences*, 17(7). 885-881.

De Floriani, L (1987) Surface representations based on triangular grids. *The Visual Computer*, 3(1): 27-50.

Falcidieno, B and M Spagnuolo (1991). A new method for the characterization of topographic surfaces. *Int. J. Geographical Information Systems*, 5(4): 397-412.

Lee, D.T. and B. Schachter (1980). Two algorithms for constructing Delaunay triangulations. *Journal of Computer and Information Science*, 9(3): 219-242.

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

Sakude, M T.S. (1992). Terrain Modeling with Bezier triangular surfaces (In Portuguese) In *Sibgrapi V*, pp. 213-222.

Shermer, T C. (1992). Recent Results in Art Galleries, *Proceedings of the IEEE*, Special Issue on Computational Geometry, 80(9). 1384-1397

Van Kreveld, M. (1996) Digital Elevation Models overview and selected TIN algorithms. Course Notes, CISM Advanced School on Algorithmic Foundations of Geographic Information Systems, Udine

A Method for Handling Data that Exhibit Mixed Spatial Variation

Bheshem Ramlal
Assistant Lecturer, Department of Surveying and Land Information, The University of the West Indies, St. Augustine, Trinidad, West Indies, bheshem@eng.uwi.tt

Kate Beard
Associate Professor, National Center for Geographic Information Analysis and Department of Spatial Information Science and Engineering, University of Maine, Orono, ME 04469. beard@spatial.maine.edu

Abstract

Global change research requires data for large expanses of the earth. Data are normally obtained by sampling at several discrete points over the area of interest. An interpolation technique is then employed on these data to estimate values for unvisited points. Most interpolation techniques assume continuous variations and data that are of the same quality. Measurement errors are assumed to be absent. These assumptions are not valid, especially because data used for global change research are often obtained from multiple sources of varying accuracy and these data represent landscapes that generally exhibit mixed spatial variation. That is, both continuous variations and abrupt changes, are present on these landscapes.

In this paper, we describe a technique that incorporates abrupt changes in the interpolation process and accommodates data from different sources that may vary in quality. We provide a description of mixed spatial variation and define each of the components of mixed variation data. Several methods have been forwarded to deal with abrupt changes. However, most of these do not adequately handle discontinuities. We forward an alternate interpolation technique based on the kriging process and provide a description of the changes required in this process to implement the technique. We also describe a method that extends the kriging process to obtain reliability estimates for values generated by this interpolation technique and discuss the advantages and limitations of the approach.

1. Introduction

Global change research requires data that encompasses large expanses of the earth. These data are normally collected at diverse scales from multiple sources that may vary significantly in reliability. However, these data are often treated as being error free and obtained from the same source. Another major assumption is that these data represent phenomena that are continuous in nature. While this may be valid for some, most of these phenomena exhibit mixed spatial variation. A phenomenon may be considered to exhibit mixed spatial variation if both continuous and discrete variations are present (Ramlal and Beard 1996). The soil landscape, for example, exhibits this behavior (Hole and Campbell 1985,

Voltz and Webster 1990). Usually these landscapes are treated as being either completely continuous with transitional variations or comprised of discrete, well defined, homogeneous units (Burrough 1993). Neither of these approaches fully capture all the variations that are present in the landscape (Burrough 1989, Heuvelink 1993). The representations provided by these models are therefore limited (Burrough 1993, Ernstrom and Lytle 1993, Ramlal and Beard 1996). Although some researchers have acknowledge the presence of discontinuities, most interpolation methods avoid the incorporation of the line of discrete change in the process (Marechal 1983, Stein et al. 1988, Voltz and Webster 1990, McBratney et al 1991, Brus 1994).

This paper describes an interpolation technique that processes data that: (i) exhibit mixed spatial variations, (ii) may have been obtained from different sources, and (iii) may vary in quality. The soil landscape is used as an example here. Note that while the discussion focuses on soil data at scales larger than global, the solution that are being forwarded may be adapted to other phenomena that exhibit global variations. The second section of this paper provides a description of mixed spatial variation and defines each of the components of mixed variation data. Several methods have been forwarded to deal with abrupt changes. The third section discusses these methods, and their advantages and limitations for handling discontinuities. The fourth section of the paper presents an alternate interpolation technique based on the kriging process and provides a description of the changes required in this process to implement the technique. This section also describes methods for incorporating discontinuities in the computation of the semi-variogram and generating estimates for unvisited points. The fifth section forwards a method to extend the kriging process to obtain reliability estimates for values generated by this interpolation technique and discusses the advantages and limitations of this approach. The last section presents a discussion of the benefits and limitations of the proposed technique.

2. The Components of Mixed Spatial Variation Data

Capturing mixed spatial variation requires the use of sampling strategies that capture both continuous variations and abrupt changes. The soil landscape cannot be completely sampled because this will destroy it (Burrough 1993). Other phenomena, such as rainfall, are not completely captured because it is financially unfeasible to obtain a comprehensive coverage of samples for large expanses of the earth. Point samples taken at intervals over the landscape suffice to provide a representation of the continuous variation that are present in phenomena that exhibit mixed spatial variations. Several sampling strategies are discussed in the literature for locating sample points for the soil landscape. See for example Burgess and Webster (1980), Van Kuilenberg *et al.* (1982), McBratney and Webster (1983), and Webster and Oliver (1992).

Phenomena that exhibit mixed spatial variations also contain discontinuities. These abrupt changes occur along lines for surfaces, and surfaces for volumes. These changes may be visible on the surface, may occur beneath the surface or may not be visible at all for some phenomena. They may be

vertical or inclined. Throughout this paper the term *line of abrupt change* is used to denote both lines and surfaces of abrupt change.

Unlike the well defined, sharp boundary used to delineate soil units, lines of abrupt change may vary from zero-width to a zone of change (Mark and Csillag 1991, Wang and Hall 1996). Attached to each side of the line of change are values for the property that exhibits abrupt change. The greatest extent to which the line of abrupt change is considered to be abrupt, rather than transitional, depends on the locational and thematic resolutions defined prior to sampling. The width of the line of change needs to be less than the locational resolution of the spatial variation being captured to be considered sharp. A change in property is considered abrupt if it significantly exceeds the thematic resolution. What is classified as significant is based on user requirements (figure 1).

The second important consideration in defining an abrupt change is the definition of a short distance. Unlike well-defined boundaries, a line of abrupt change may vary in width from zero-width to a number of meters or hundreds of meters depending on the phenomenon being mapped (figure 1). What is classified as a short distance may vary considerably depending on the expected application of the data and the property being studied. For abrupt spatial changes, the resolution or scale at which they are identifiable needs to be included with these data.

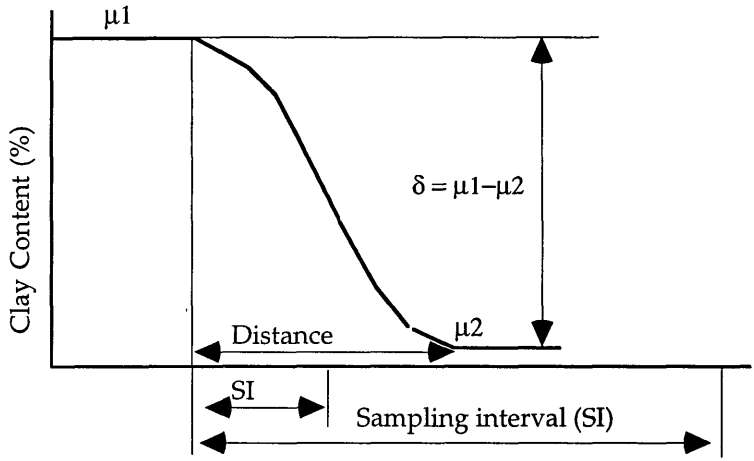


Figure 1. A short distance is determined by the ratio between the distance over which the significant variation occurs and the sampling interval. A tolerance ratio may be set before lines of abrupt change are delineated

The variation of a property may be considered to be abrupt if

$$\delta < Ave \text{ and } \frac{d}{SI} < R \qquad (1)$$

Where δ is the established significant variation for a property,
Ave is the average change in variation for distance , d

SI is the sampling interval for point samples, and
R is an established ratio of distance to sampling interval that is considered short. Note that d and R are established after the reconnaissance survey but before the final survey is undertaken.

3. Existing Approaches for Handling Discontinuities

Several methods have been forwarded to deal with discontinuities: stratification (McBratney et al. 1991), the combination of soil classification and kriging (Stein et al. 1988a), within class kriging (Voltz and Webster 1990), and spline interpolation (Voltz and Webster 1990).

McBratney *et al.* (1991) conducted a study to determine the effects of discontinuities on the estimates of unvisited points. They measured the clay content for an area where a discontinuity was identifiable on the terrain. Values for unvisited points were computed using kriging for two scenarios. The first assumed the absence of any discontinuities. The second scenario took the discontinuity into consideration. In the presence of a discontinuity, the region was split into two sub-regions and the point samples were separated according to their location. McBratney *et al.* (1991) showed that stratification provided slightly better results than non-stratified interpolation. However, problems arose because of edge effects and the fact that less points were used in the interpolation of values within each region (McBratney et al. 1991, Brus 1994). The discontinuity itself was not used in estimating values for points located close to it. This led to a reduction of the reliability of attributes along the boundary.

In an attempt to accommodate both discrete and continuous variations in the soil landscape, several researchers have combined data from soil maps and point sampling (Stein et al. 1988, Voltz and Webster 1990, Heuvelink and Bierkens 1992). The study conducted by Heuvelink and Bierkens is described here. In this approach, points were stratified to ensure that only points within a mapping unit were used to estimate values within the unit. Each unit was kriged separately. The resulting values from kriging within these units and the values from soil map predictions were combined by taking a weighted average of these values. The results showed that combining data from these two methods produced a more accurate map than when either the kriging method or soil maps was used separately. In a similar study, Voltz and Webster (1990) came to this conclusion as well. Both studies demonstrated that this method suffers two major disadvantages (Heuvelink and Bierkens 1992, Voltz and Webster 1990): (1) it assumes that all units are separated by sharp boundaries, and (2) the number of points available for the computation of variograms are reduced.

Voltz and Webster (1990) have forwarded a method that addresses the second disadvantage of the above method. Instead of stratifying points using soil boundaries, they stratified using soil classes. Points that are located in all mapping units of the same class were grouped together. These points were used to compute variograms for each class. Kriging was carried out for each mapping unit using these variograms. The results from this process showed an

improvement over the use of simple kriging or the use of soil maps separately (Voltz and Webster 1990). However, this approach suffers many disadvantages. Homogeneity within soil classes was assumed. This is not necessarily valid (Voltz and Webster 1990). Additionally, sharp boundaries and homogeneous mapping units were assumed. These assumptions do not always hold. Another problem with this method is that the spatial context is lost when points are grouped together based on soil classes. Both this and the previous method do not address the problem of edge effects that occur at the boundaries of mapping units.

Voltz and Webster (1990) also examined the use of spline interpolation for handling discontinuities. They chose joining points for spline curves, called knots, such that local variations, especially soil boundaries, will not be smoothed out. The results of this method were compared with the results from within class kriging, soil classification, and stratification using soil mapping units. Spline interpolation performed better than traditional soil classification. However, both stratification and within class kriging performed better than spline interpolation. Voltz and Webster (1990) concluded that the major limitation of this approach is that it over-accentuates boundaries and fluctuates more than kriging curves, which implied too much influence from local variations.

4. An Alternate Approach

Unlike the methods forwarded by Stein *et al.* (1988) and Voltz and Webster (1990) to handle discontinuities, there are no soil mapping units and soil classes in the mixed variation model. These two approaches are not appropriate here. The method presented by McBratney *et al.* (1991) assumes a single discontinuity that spanned the entire length of the area. Two separate areas were therefore easily identifiable. In the mixed variation model, more than one line of abrupt change is present. As a consequence, an alternate solution is required. An alternate method that accommodates discontinuities and generates reliability estimates for interpolated points is presented. This method is based on the kriging process. Discussions of kriging may be found in Cressie (1991), Burrough (1993), Burgess and Webster (1980), and Davis (1991).

4.1 Generating Semi-Variograms

The inclusion of values from lines of abrupt change in the computation of semi- variograms requires the use of points on these lines rather than the lines themselves. Points are used instead of lines so that over-influence on the resulting semi-variograms, by these lines, is avoided. The following procedure is forwarded to achieve this. The assumption here is that lines of abrupt change are used to stratify points into regions and that these lines will influence the range of the semi-variogram. Since discontinuities may not be present in all directions, separate semi-variograms should be computed for different directions. This may be done for eight directions (Webster and Oliver 1990).

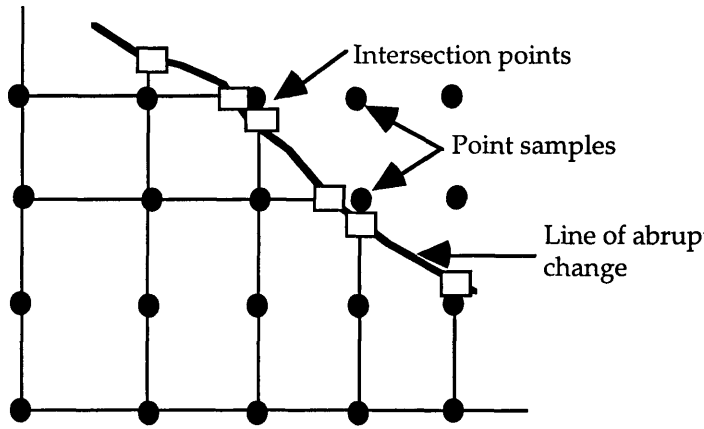


Figure 2. Method for choosing points on the line of change for computing the semi-variogram

If four directions are used, semi-variograms may be generated as follows. Points on the discontinuity are chosen by extending lines horizontally and vertically to intersect the line of abrupt change (figure 2). Intersection points from the respective directions are included in the computation of the semi-variogram for each direction. These semi-variograms can then be used to determine the ranges in the respective directions.

4.2 Including Discontinuities in the Interpolation Process

To accommodate discontinuities in the interpolation process, the following procedure is forwarded. This method follows a combination of strategies proposed by Buys et al. (1991) and Chen (1988) for including break lines in digital elevation models. Refer to figure 3. The steps in the process are as follows: (1) Identify the points that fall within the range for point A. The number of points chosen will depend on the range, which may be obtained from the semi-variogram for these observations; (2) Search for points that fall on the opposite side of the boundary from point A. (3) Join point A to each of these points using a straight line. (4) Find the intersection points between A and each of these points and the line of abrupt change. Intersection points are shown as rectangles in the figure. This method uses points located on the line of abrupt change instead of points located on the other side of the boundary. Note that this method ensures that the number of intersection points is always the same as the number of points omitted.

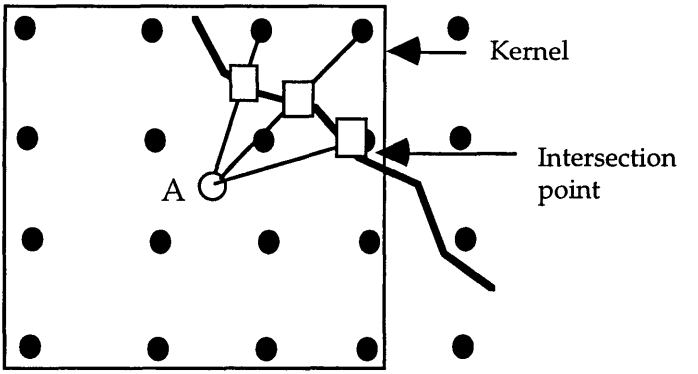


Figure 3. Finding intersection points on the line of abrupt change

Once the positions of these points are computed, the next step is to determine an attribute value for each of them. This will depend on the data stored for the discontinuity. If a value is stored for the line, then this is assigned to each point. If no value exists, then separate values need to be computed. This may be computed using the points within the kernel on the same side of the line of abrupt change as point A. Edge effects may occur because of the absence of values at the discontinuity.

5. Estimating Reliability

An example using forty-two points to estimate the value of thirty-two unvisited points is given below to demonstrate how (1) lines of abrupt change may be incorporated into the kriging process and (2) positional and attribute accuracy may be propagated to obtain a single reliability measure for interpolated points. A linear semi-variogram is assumed for this data. Figure 4. shows point samples and the grid of unvisited points to be estimated.

The attribute values represent percent Ash content in the soil. The line of abrupt change was arbitrarily placed in position and assigned two attribute values obtained by averaging values of point samples on the left and on the right of the line. The positional and attribute errors were assumed to be half the least unit of measurement. It is also assumed that the spatial variation is isotropic and that sample points closer to the unvisited points are more influential than points further away. While these assumptions are not always valid, they are more often true than any other situation (Webster and McBratney 1983a). A normal distribution is assumed for the behavior of errors in attributes and positions. This assumption is not totally valid but has been used many times before and is an accepted one (Heuvelink 1993). An interval of 0.25 units between unvisited points is used here. The choice of cell size depended on the sampling interval only. This is not always the most appropriate approach.

To show how the estimated value for unvisited points will vary with the introduction of a line of abrupt change and with different values of position and attributes, a program was written to interpolate values for 32 points using punctual kriging. This program executes a simulation process that randomly distorts position and attribute values of sample points and lines of abrupt change while maintaining the topological relationships between points and the line, to compute multiple estimates for unvisited points. Distortions are based on the positional and attribute error of the data. Values may be generated for different ranges of confidence interval. A distribution of these estimates may be studied to determine confidence intervals for unvisited points.

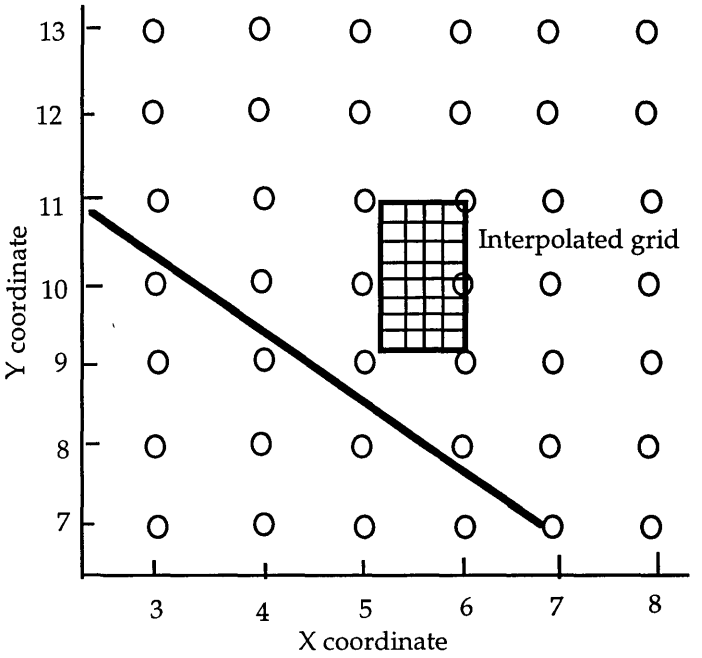


Figure 4. Point samples (7 x 6) and line of abrupt change used to estimate values for the grid (8 x 4)

Estimates were obtained for two scenarios. In the first, only point samples were used. The second scenario used both point samples and a line of abrupt change. It was assumed that all points and the line were of the same positional and attribute accuracy. However, this may be easily changed to accommodate data of varying levels of accuracies. The ranges of estimates about the mean for each point for each scenario are shown in figures 5 and 6. The mean values for each unvisited point (not shown in the diagrams below) in the two scenarios were the same except for the estimates of points that were influenced by the line. This result demonstrate that the points blocked off by the line were not used instead values on line were utilized.

The range of estimates presented in figures 5. and 6. incorporate both the positional and attribute error that were present in the original data. These ranges can be used as a measure of the reliability for the estimates. This method therefore combines these two components of quality to obtain a single measure of reliability. The results of interpolating in the presence of a line of the same accuracy as the sample points should be the same or better than the use of points only. Comparing figures 5 and 6, it may be noted that the range of values are very similar. In fact, the line generates a lower range of variations at unvisited points closer to it. A statistical significance test at the 99% confidence level supports this conclusion.

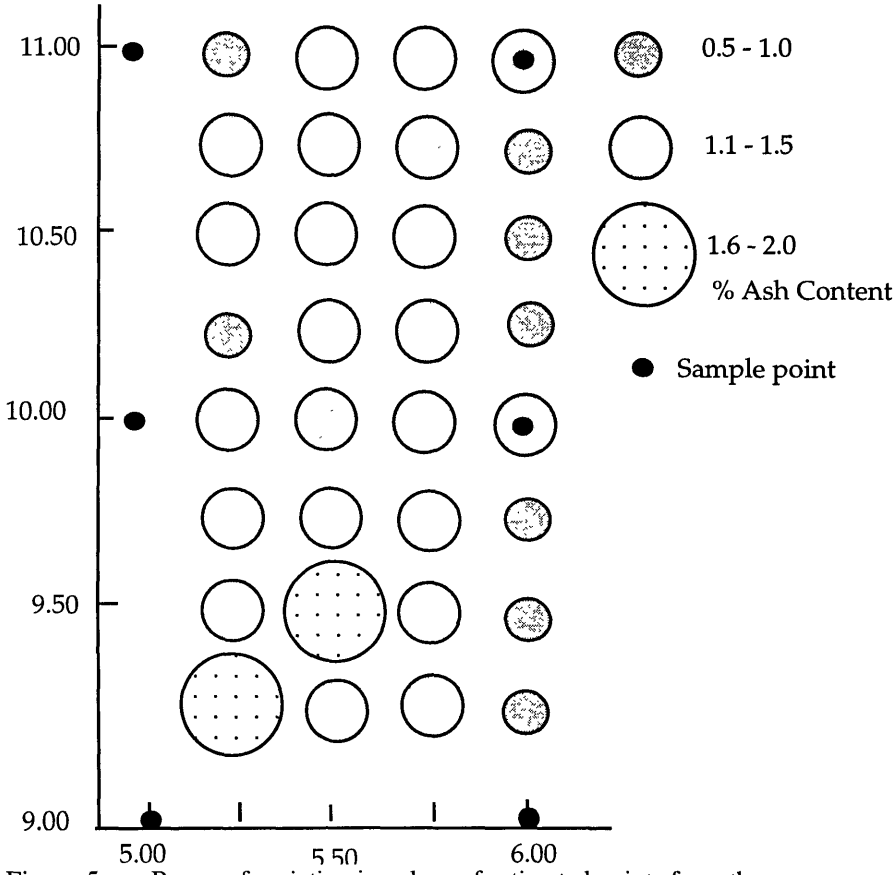


Figure 5. Range of variation in values of estimated points from the mean given 99 percent confidence limits in position and attribute accuracy (values represent percent Ash content in soil)

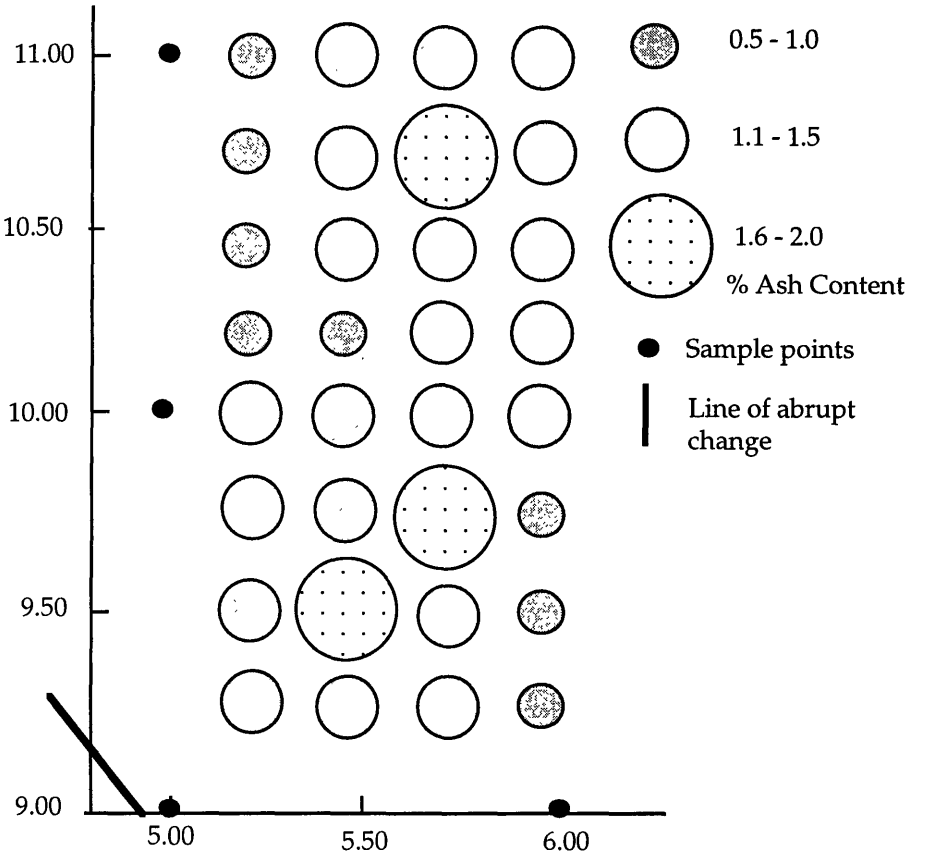


Figure 6. Range in estimate when points and line of the same accuracy in position and attribute are used.

5. Discussion and Conclusion

Unlike other methods of dealing with mixed variation data, the approach presented in this paper fully incorporates discontinuities in the interpolation process. This ensures a higher reliability of estimates for unvisited points that are located close to the line of abrupt change. Edge effects are therefore reduced. The line of abrupt change is incorporated such that its importance is not over-emphasized and its influence is controlled by the same weighting process used for points only in the kriging process. Additionally, this method combines positional and attribute errors from the original data, to provide a single reliability measure for estimates. This method is therefore ideally suited for handling data that exhibit mixed spatial variations.

References

Brus, D.J. (1994). "Improving design-based estimation of spatial means by soil map stratification. A case study of phosphate saturation." Geoderma 62: 233-246.

Burgess, T.M. and Webster, R. (1980a). "Optimal interpolation and isarithmic mapping of soil properties I. The semi-variogram and punctual Kriging." Journal of Soil Science 31: 315-331.

Burgess, T.M. Webster, R. McBratney, A.B. (1981) "Optimal interpolation and isarithmic mapping of soil properties IV. Sampling Strategy." Journal of Soil Science 32:643-659.

Burrough, P.A. (1993). "Soil variability: a late 20th century view." Soil and Fertilizers 56.5: 529-562.

Buys, J. Messerschmidt, H.J. Botha, J.F. (1991). "Including known discontinuities directly into a triangular irregular mesh for automatic contouring purposes." Computers and Geosciences 17.7: 875-881.

Chen, Z. (1988). Break lines on terrain surface. GIS/LIS'88.

Cressie, N. (1993) Statistics for Spatial Data (Revised Edition) Toronto: Wiley and Sons.

Davis, J.C. (1991). Statistics and Data Analysis in Geology (2nd Edition). New York: Wiley and Sons.

Ernstrom, D. J. and Lytle, D. (1993). "Enhanced soil information systems from advances in computer technology." Geoderma 60: 327-341.

Heuvelink, G.B.M. and Bierkens, M.F.P. (1992). "Combining soil maps with interpolations from point observations to predict quantitative soil properties." Geoderma 55: 1-15.

Heuvelink, G.B.M. (1993). Error propagation in quantitative spatial modelling: application in geographical information systems Published PhD. Thesis, University of Utrecht, The Netherlands.

Heuvelink, G.B.M. and Burrough, P.A. (1993). "Error propagation in cartographic modelling using Boolean logic and continuous classification." Int. J. Geographical Information Systems 7.3: 231-246.

Hole, F.D. and Campbell, J.B. (1985). Soil Landscape Analysis. New Jersey: Rowman and Allanheld.

Mark, D.M. and Csillag, F. (1991). "The Nature of Boundaries on "Area-Class" Maps." Cartographica 26: 65-78.

McBratney, A.B. Hart. G.A. McGarry, D. (1991). "The use of partitioning to improve the representation of geostatistically mapped soil attributes." Journal of Soil Science 42: 513-532.

McBratney, A.B. and Webster, R. (1983a). "How many observations are needed for regional estimation of soil properties." Journal of Soil Science 38.3: 177-183.

McBratney, A.B. and Webster, R. (1983b). "Optimal interpolation and isarithmic mapping of soil properties V. Co-regionalization and multiple sampling strategy." Journal of Soil Science 34: 137-162.

Ramlal, B. and Beard, K. (1996). " An alternate paradigm for the representing mixed spatial variations." Third NCGIA conference on Environmental Modeling and GIS, Santa Fe, New Mexico.

Stein, A. Hoogerwerf, M. Bouma, J. (1988). "Use of soil-map delineations to improve (co-)kriging of point data on moisture deficits." Geoderma 43: 163-177.

Van Kuilenburg, J. De Gruijter, J.J. Marsman, B.A. Bouma, J. (1982). "Accuracy of spatial interpolation between point data on soil moisture supply capacity, compared with estimates from mapping units." Geoderma 27: 311-325.

Voltz, M. and Webster, R. (1990). "A comparison of kriging, cubic splines and classification for predicting soil properties from sample information." Journal of Soil Science 41: 473-490.

Wang, F. and Brent Hall, G. (1996). " Fuzzy representation of geographical boundaries in GIS." IJGIS Vol. 10 No. 5: 573-590

Webster, R. and Oliver, M.A. (1990). Statistical Methods in Soil and Land Resources Survey. Oxford: Oxford University Press.

Webster, R. and Oliver, M.A. (1992). "Sample adequately to estimate variograms of soil properties." Journal of Soil Science 43: 177-192.

DEMOGRAPHY IN GLOBAL CHANGE STUDIES

Waldo Tobler
Professor Emeritus
Geography Department
University of California
Santa Barbara, CA 93106-4060
Voice: (805) 964-0116
Email: tobler@geog.ucsb.edu
Fax: (805) 893-3146

ABSTRACT

Demographic information is usually provided on a national basis, but we know that countries are ephemeral phenomena. As an alternate scheme one might use ecological zones rather than nation states to organize environmental data. But there is no agreement as to what these zones should be. By way of contrast, global environmental studies using satellites as collection instruments yield results indexed by latitude and longitude. Thus it makes some sense to assemble information on the terrestrial arrangement of people in a compatible manner. This alternative is explored in the work described here, in which latitude/longitude quadrilaterals are used as bins for population information. This data format also has considerable advantage for analytical studies in which spatial series can be thought of as a two dimensional extension of time series, or for simulation modeling, etc. The result of our recent work is a five minute by five minute raster of estimated 1994 world population generated from over nineteen thousand administrative unit boundaries and covering 221 countries. The number of people in the total countries is estimated to be 5.618×10^9, spread over 1.32×10^8 square kilometers of land. The full report details the methods used, problems encountered, applicability to urban areas, movement modeling and other uses, and needed extensions.

DESCRIPTION

The Global Demography Project at the National Center for Geographic Information and Analysis in the Geography Depatment of the University of California at Santa Barbara recently completed an inventory of world population by 5 minute quadrilaterals of latitude and longitude. The choice of 5 minute quadrilaterls is intended to be compatible with other global data compilations, and is intended for medium resolution studies not detailed local invesitgations. The detailed report is available from NCGIA* and the data are available on FTP from CIESIN**. The project involved the assembly of population estimates from 19,032 administrative units (defined by polygons using latitude, longitude coordinates) from countries of the world. Since the census dates vary, from 1979 to 1994, we to extrapolated the global number of people to a common date; this was chosen to be 1994. The 5 minute raster, 1548 rows (57S to 72N latitude) by 4320 columns (360 East-West degrees) in size covers 9.27 by 9.27 km (85.8 km^2) at the equator to 3.1 by 9 27 km (28.7 km^2) at the northen limit Table I). The quadrilaterals at the southern limit are approximately 49 km^2 in area. These numbers define the resolution, and, from the sampling theorem, also the minimal patterns which can be detected at each latitude. The resolution is thus 12 times that of the previously available data, and a decade more current (Matthews, 1983). A polygon to raster program reassigns the population numbers from the sub-national administrative unit polygons to the spherical quadrilaterals. This is followed by a pycnophylactic areal smoothing, described elsewhere (Tobler, 1996), to yield a more realistic

*The complete report by W. Tobler, U. Deichmann, J. Gottsegen, & K. Maloy, "The Global Demography Project", Technical Report 95-6, NCGIA Geography Department, University of California, 75pp + PC diskette, is available from NCGIA Publications, Geography Department, University of California, Santa Barbara, 93106-4060. Or contact NCGIAPUB@NCGIA.UCSB.EDU

**CIESIN can be contacted at Info@CIESIN.org

TABLE I

Difference in quadrilateral area using the exact ellipsoidal formula and using a mean radius sphere, in square kilometers. Based on a quadrangle area of 5 minutes in latitude and 5 minutes in longitude.

Latitude (Degrees)	Ellipsoid Area	Spherical Area	Diff. Sq km	Percent Diff.	E-W km Distance
0.00	85 480	85.863	-.384	-.449	9.266
5.00	85.158	85.531	-.374	-.439	9.230
10.00	84.204	84.548	-.344	-.408	9.124
15.00	82.625	82.922	-.296	-.358	8.949
20.00	80.430	80.664	-.234	-.291	8.705
25.00	77.631	77.792	-.162	-.208	8.395
30.00	74.245	74.329	-.083	-.112	8 021
35.00	70.295	70.299	-.004	-.006	7 587
40.00	65.805	65.735	0.070	0.107	7.094
45.00	60.806	60.670	0.136	0.223	6.547
50.00	55.332	55.114	0.188	0.340	5.951
55 00	49.422	49.198	0.224	0.452	5.309
60.00	43.118	42.878	0.241	0.558	4.627
65.00	36.469	36.231	0.238	0.653	3.910
70.00	29.526	29.308	0.127	0.736	3.163
75.00	22.342	22.163	0.179	0.802	2.392
80.00	14.976	14.849	0.128	0.851	1.602
85.00	7.487	7.421	0.066	0.881	0.801

Latitude refers to the southwest corner of the 5′ by 5′ quadrangle. The radius of 6371.007178 kilometers gives a spherical surface area of 510,065,621 km^2 equal to that of the WGS 1984 ellipsoid. Authalic latitude has not been used.

population distribution. An additional product is a tabulation of sub-national administrative unit names by 5 minutes in latitude and longitude. This can be used to assign other socio-economic variables to the spherical raster without significant computational cost. The centroids of the 19,032 polygons are also known by coodinates, and are given in the full report. Proprietary restrictions by a few countries prohibit our distribution of some of the sub-national boundary coodinates. The initial raster sizes required nearly 47 Mb of storage, but, since much of the earth's surface contains virtually no people this could be compressed to just over 5 Mb.

The accompanying viewgraphs illustrate the results for several regions, either by showing the boundary polygons used or by centroid locations or by displaying some of the resulting population distributions.

ANALYSIS

The data lend themselves to several kinds of analysis. The raster format has considerable advantage for analytical studies in which spatial series can be thought of as a two dimensional extension of time series, or for simulation modeling, etc. Population density, for example, can be defined using the finite difference approximation to the gradient of the population given by the grid. Only two simple results are presnted here: The population distribution by latitude, and by longitude, using a data aggregation to 30 minutes is shown on viewgraphs. The latitudinal distribution of people is *a peu pres* unimodal with a peak around 30 degrees north. The longitudinal distribution shows 4 peaks, one for the Americas (120W to 30W), one bimodal bump representing Europe and Africa (7W to 52E), with the two largest peaks for India (67E to 90E) and China-East Asia (102E to 135E). A second simple compution gives the total number of people near water areas, by 30 minutes of latitude and by 30 minutes of longitude (also on view graphs). The cumulative number of people adjacent to water, for example is 465 million, using the 4 neighboring 30 minute cells, and 817 million using an 8 neighbor criterion. These numbers represent 8.3% and 14.5%, of the total world population The use of 30 minute data rather than the 5 minute data is because these results are more easily comprehended and because this size easily fits within a DOS program on a PC. The computation is simplified because water is represnted in the raster by a negative number.

CITATIONS

Matthews, E. (1983) "Global Vegetation and Land Use: New High-Resolution Data Bases for Climate Studies", *J. Climate & Applied Meteorology*, 22:474-487.

Tobler, W. (1996) "Converting Administrative Data to a Continuous Field on a Sphere", in *Proceedings, Third International Confrence/Workshop on Integrating GIS and Environmental Modeling*, Santa Fe NM, Jan 21-26, 1996. NCGIA compact disk, Santa Barbara.

INTERPOLATION OVER LARGE DISTANCES USING SPHEREKIT

Robert Raskin, Associate Scientist, Applied Research Corporation,
Jet Propulsion Laboratory Mail Code 525-389, Pasadena, CA 91109 USA

Chris Funk, Graduate Assistant, Department of Geography,
University of California, Santa Barbara, CA 93106 USA

Cort Willmott, Professor, Department of Geography, University of Delaware,
Newark, DE 19716 USA

ABSTRACT

Spherekit is a spatial interpolation toolkit developed and distributed over the internet by the National Center for Geographic Information and Analysis (NCGIA). A unique feature of the software is its ability to work directly with the spherical geometry of the earth. Thus, distances, areas, and directions are spherically based, and interpolation can be carried out over large distances without distortions induced by the use of planar projections. The user can select from several interpolation methods that have been adapted to the sphere. The package also features "smart interpolation" capabilities to incorporate knowledge of the underlying physical processes that produced some of the spatial variability. Error analysis using cross-validation is built-in to compare the relative performance of interpolation algorithms or parameter settings. The cross-validation errors can themselves be interpolated to a uniform grid to reduce spatial bias. The capabilities of Spherekit are demonstrated using three examples.

OVERVIEW

Spherekit is a spatial interpolation software toolkit developed at NCGIA as part of Initiative 15 (Multiple Roles of GIS in Global Change Research). The source code is available over the internet without charge to the user. The package features several unique capabilities.

Spherekit permits interpolation over continental or global scales because its computations are based upon spherical distances and orientations (Raskin, 1994). Conventional interpolations (Watson, 1992) are based upon planar projections of the earth that produce distortions of some kind over large distances. In Spherekit, projections are applied only for display purposes after

the interpolation has been carried out in spherical geometry. The user can select from several interpolation algorithms that have been adapted to the sphere: inverse distance weighting, thin plate splines, multiquadrics, triangulation, and kriging.

Spherekit permits the user to incorporate knowledge or information about the processes that produced the underlying spatial variations. A built-in equation editor and a collection of nonlinear transforms allows the user to create and experiment with new, physically meaningful variables from the independent and dependent variables available. This "smart" interpolation capability allows Spherekit to intelligently interpolate using auxiliary information. A digital elevation model (DEM) is included with the package. One use of the smart interpolation feature is to incorporate elevation information when interpolating variables that are correlated with height.

Error analysis is an integrated component of Spherekit. This makes the package particularly useful for comparing interpolation methods and parameters. The performance of a method is measured using cross-validation. The cross-validation error is defined at each observation point as the difference between its actual value and its estimated interpolated value using the remaining n-1 points. The resulting error field can be displayed either at the data points or interpolated to a regular grid to reduce spatial biases. Error difference fields, comparing a pair of methods and/or parameter settings, can be easily created and displayed.

Spherekit helps the user manage the various files that have been read in or created. The file management window for a sample session is shown in Figure 1. The example shows file listings for observation data, grids, networks, interpolation methods, interpolation results, error fields, and derived variables. Clicking on any field name displays all known metadata for that field. This window also serves as the Spherekit main menu; six main menu options appear along the top of the window.

INTERPOLATION METHODS

Spherekit was designed to be usable by researchers analyzing global scale datasets. Several standard interpolation methods have been modified for use on the sphere by utilizing spherical distance in place of Euclidean distance. Additional modifications for sphericity are used where possible. Most of the interpolation methods can be implemented as either global or local methods. For local methods, a neighborhood size is specified either as a radius, the number of included points, or an average number of points. Overrides are available to bound these values, if desired. Five interpolation methods are available:

Figure 1 Example of the file management window

* Inverse distance weighting
* Multiquadric
* Thin plate spline
* Kriging
* Triangulation

For the inverse distance weighting method, the user can choose from three weighting functions: inverse power, Shepard (1968), and a smoothing function. The user also can select levels of anisotropy and gradient correction (Shepard, 1968). The bias correction deweights clustered points to reduce spatial bias. The gradient correction permits extremum values to occur at locations other than the observation points.

The multiquadric and spline methods involve inversion of an n×n matrix for n data points. For n larger than several hundred, the user should specify a neighborhood for carrying out local fits. This will invert smaller matrices for each interpolation point rather than performing a single large matrix inversion over the entire domain. The user is warned to use a local fit if the storage requirements for carrying out the n×n inversion exceed the space available.

The generalization of multiquadric method to the sphere has been formulated by Pottmann and Eck (1990). The thin plate spline implementation is a spherical extension of the methods described in Franke (1982).

The kriging implementation is that of kriging with a trend (universal kriging). Semivariograms are computed using the GSLIB library software package (Deutsch and Journel, 1992). Exponential, Gaussian, spherical, and linear models are supported, all using spherical distance. A summation of two of these models is permitted. The semivariogram results also can be used for exploratory analysis purposes; an example is provided in the next section.

The triangulation method uses the Delaunay triangulation to identify the nearby observation points to be used in the interpolation. Renka's spherical algorithm (Renka, 1984) is used to carry out the interpolation. The user can choose either a linear interpolation of the values at the triangle vertices or a polynomial fit, obtained by performing an initial cubic spline fit along the edges.

APPLICATIONS

Smart interpolation

"Smart" interpolation improves the performance of traditional interpolations by using knowledge of the processes that produced the spatial variations (Willmott and Matsuura, 1995). In this example, we use the physical law that temperature falls off with altitude, roughly at the environmental lapse rate. Standard and topologically aided interpolations are compared using a sparse network of 160 weather stations in China. The data set is deficient in that high altitude locations in the Himalayan mountains are underrepresented. Figure 2 shows the interpolated temperature field (in °C) using the multiquadric method. The fit based on the sparse dataset fails to take into account the large variations in topography that produce very low temperatures at high altitudes.

Figure 3 shows the corresponding "smart" interpolation based on an interpolation of the derived variable "sea level temperature." In this example, the first-order effect of the temperature- elevation relation has been incorporated into the interpolation. That is, the "smart" interpolation captures the climatological influences of topography. The low temperatures associated with the mountains of western China are now visible, despite the lack of high altitude temperature stations.

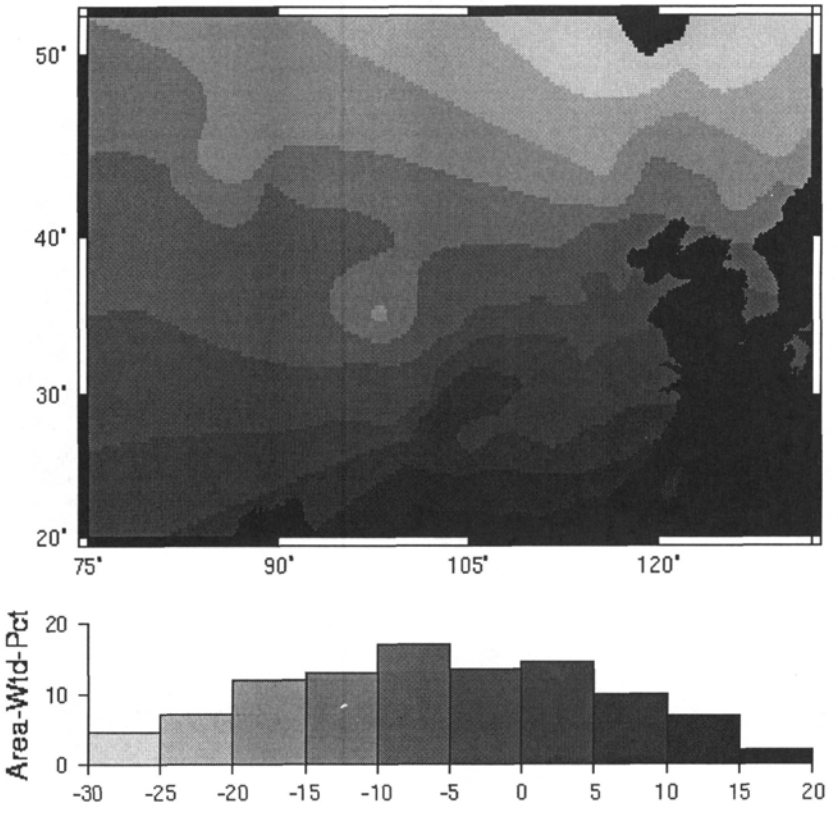


Figure 2 Conventional interpolation (°C)

The interpolation was performed using the following steps:

1. Reduce temperatures to sea level using the environmental lapse rate (SeaLevTemp= Temp + EnvLapseRate * Elevation)

2. Interpolate the "sea level" temperatures to a one-degree grid using the multiquadric method

3. Reintroduce elevation effect on the interpolated field (Temp= SeaLevelTemp - EnvLapseRate * Elevation)

This final step (the inversion of the operations inherent in Step 1) is carried out automatically by Spherekit. The user does not have to explicitly return the sea-level temperatures to actual temperatures.

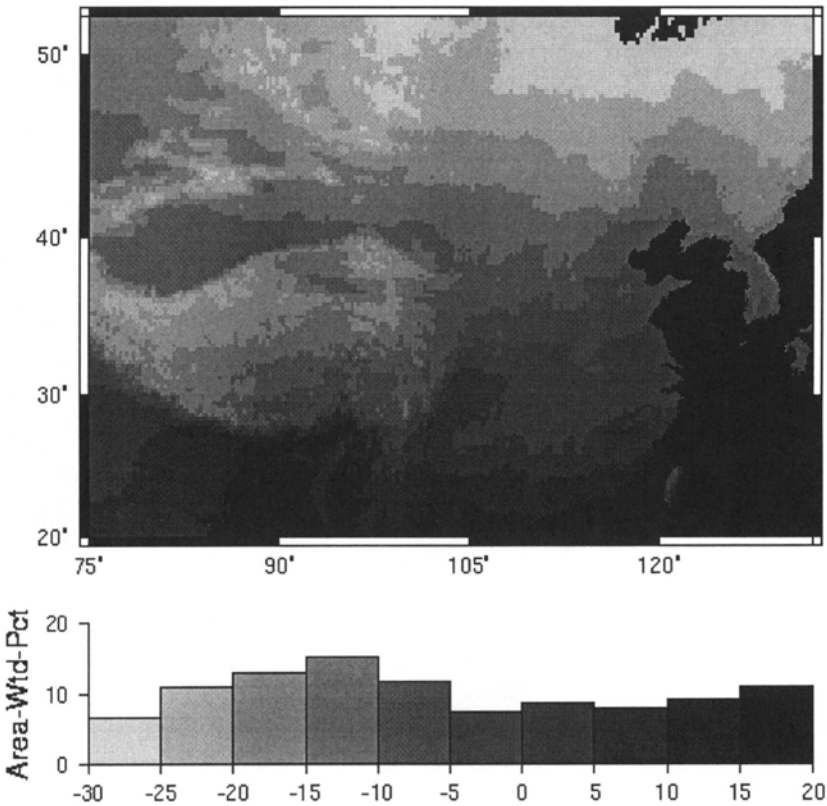
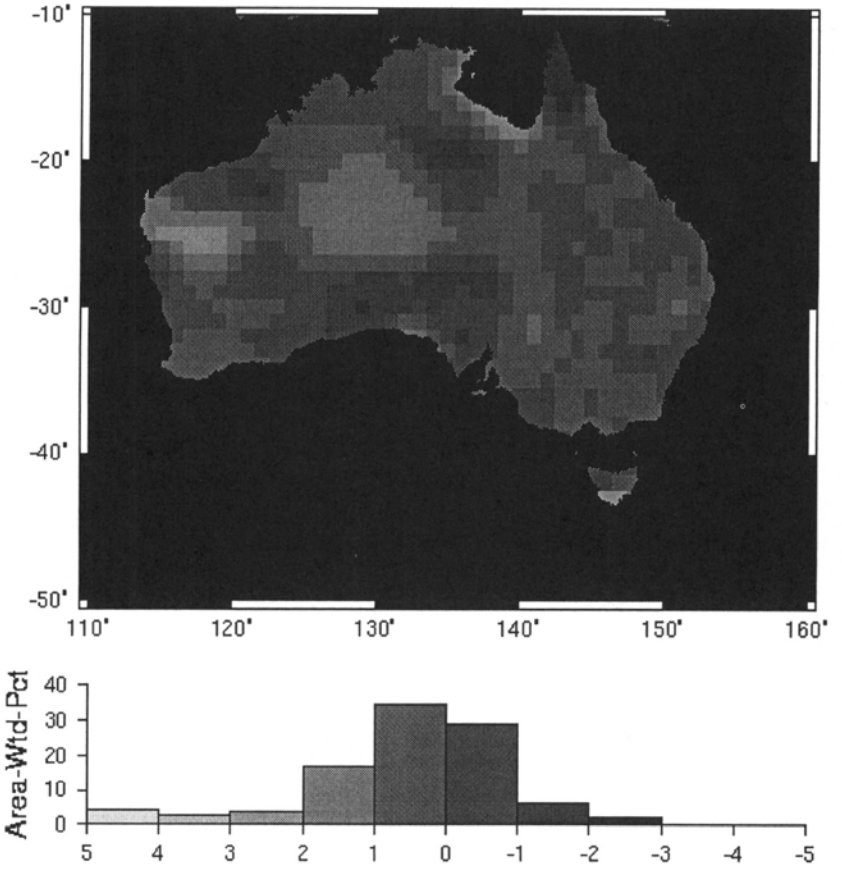


Figure 3 Smart interpolation (°C)

Error analysis

This example demonstrates the cross-validation analysis capabilities of Spherekit. In cross-validation, a data point is removed and its value is interpolated using the remaining n-1 points. The difference (actual - predicted) is the interpolation error at that point. Spherekit provides the option of interpolating the errors to a regular grid to reduce the spatial bias. Figure 4 shows the cross-validation error of a temperature dataset for Australia. Thin-plate splines were used as the interpolation method; the gridded plot reveals the one-degree granularity of the interpolation. Errors are reported in terms of three measures: mean average, mean bias, and root mean square.



MAE	0.593	(Mean average error)
MBE	-0.13	(Mean bias error)
RMSE	0.873	(Root mean square error)
MIN	-4.568	
MAX	3.173	
COUNT	1,248	

Figure 4 Thin-plate spline cross-validation errors (°C)

Spatial variability

A final example demonstrates the exploratory analysis capabilities of the package. A global temperature dataset is used to demonstrate long-distance correlations present in climate data. As Spherekit computes distances using

great circle distances, distances at continental and global scales are computed correctly.

Figure 5 shows an isotropic semivariogram of the dataset. There is a plateau in the semivariogram in the 2000-4000 km range and a sharp rise thereafter. This calculation is repeated using anisotropic semivariograms in the east-west and north-south directions. Figure 6 (the east-west semivariogram) displays the plateau more prominently. This characteristic corresponds to the common notion that zonal variations are relatively small. The north-south variations in Figure 7 vary at shorter distances, as would be expected. Interestingly, the semivariogram falls after reaching a peak; presumably this is due to a return to the same latitude zone at these distances.

AVAILABILITY

Spherekit runs on most UNIX-based machines. The source code can be downloaded from the Spherekit home page at:

www.ncgia.ucsb.edu/pubs/spherekit/main.html

The code uses Tcl/Tk for its Graphical User Interface (GUI), Generic Mapping Tools (GMT) for display of output fields, and netCDF for storing the DEM data. All of these auxiliary packages are required and can be downloaded together with Spherekit.

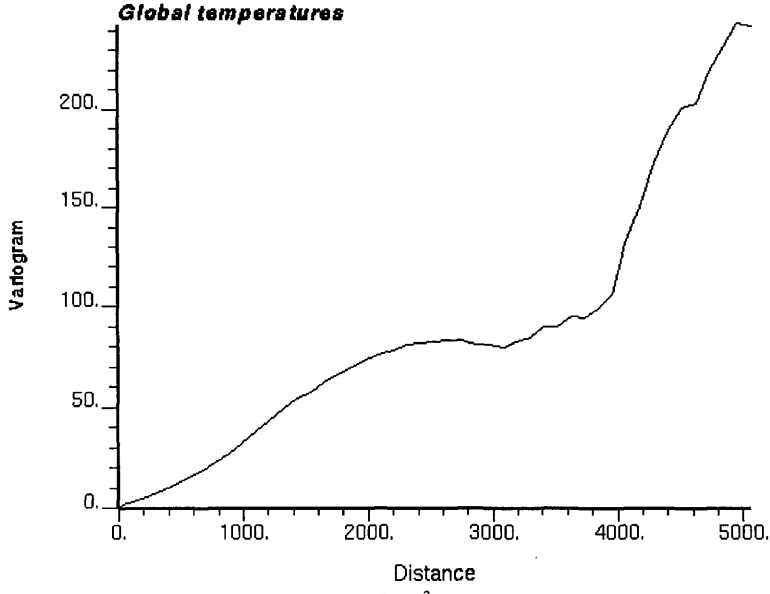


Figure 5 Isotropic semivariogram $(°C)^2$

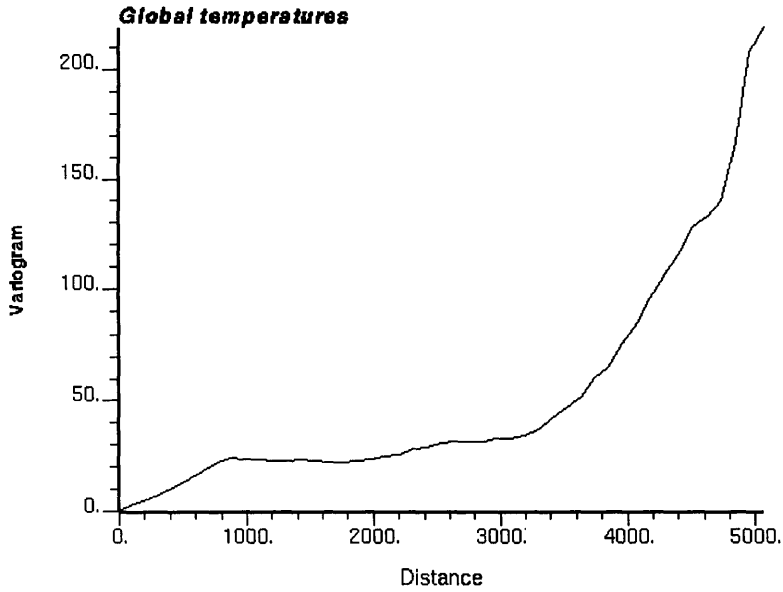


Figure 6 East-West semivariogram (°C)^2

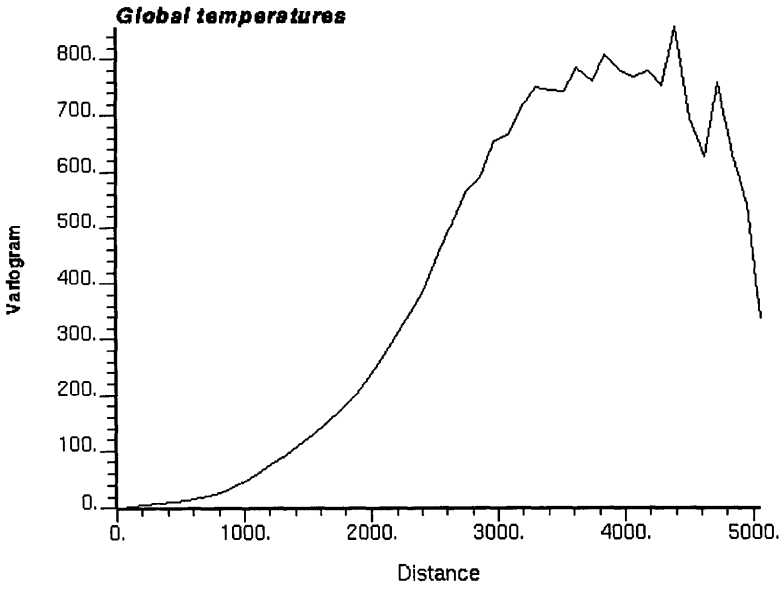


Figure 7 North-South semivariogram (°C)^2

REFERENCES

Deutsch, C. V. and A. G. Journel (1992). *GSLIB: Geostatistical Software Library and User's Guide*, New York, Oxford University Press.

Franke, R. (1982). Scattered data interpolation: Tests of some methods, *Math. Comp.*, 46: 181-200.

Pottmann, H. and M. Eck (1990). Modified multiquadric methods for scattered data interpolation over a sphere, *Computer Aided Design*, 7: 313-321.

Raskin, R. G. (1994). Spatial analysis on the sphere, Technical Report 94-7, National Center for Geographic Information and Analysis, 44 pp.

Renka, R. J. (1984). Interpolation of data on the surface of a sphere, *ACM Transactions on Mathematical Software*, 10: 417-436.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data, Proc. 23rd National Conference ACM, ACM, pp. 517-524.

Watson, D. F. (1992). *Contouring: A Guide to the Analysis and Display of Spatial Data*, Pergamon Press, 321 pp.

Willmott, C.J. and K. Matsuura (1995). Smart interpolation of annually averaged air temperature in the United States, *Journal of Applied Meteorology*, 34(12), 2577-2586.

WILL CONCERN FOR EQUITY BE A DEFINING TREND IN LIS/GIS IN THE NEXT DECADE?

David L. Tulloch, Research Assistant,
Bernard J. Niemann, Jr., Professor and Director,
Land Information and Computer Graphics Facility,
University of Wisconsin-Madison, and
Earl F. Epstein, Professor,
School of Natural Resources, Ohio State University

ABSTRACT

This paper presents looks at the role that "equity" could play in the future of multipurpose land information systems (MPLIS) development. We propose a model of MPLIS development that we feel conveys new ideas about the role of "equity" in community systems development. Additionally, evidence is provided, both anecdotal experiences and survey results, which contributes to this discussion. Finally, we discuss what this all may mean in the future for communities seeking to develop MPLIS.

INTRODUCTION

Whether directly or indirectly, a large portion of GIS research addresses the increased efficiency and effectiveness of systems within organizations. Much less concern has been placed on broadening the view of these systems to the community-wide context in which they exist. Within the community-wide context, a system's value is not simply measured by the benefits accruing to the operating organization -- efficiency and effectiveness -- but also by those benefits enjoyed by the community -- equity. Consideration of this broader context with its added benefits stream suggests that MPLIS development has not completed its full-life-cycle until the system reaches a final stage of development that we call democratization, characterized by broader community-wide participation in land management. Inherent in this concept is that a MPLIS that does not achieve democratization (i.e. the system produces outcomes of improved equity) should not be considered fully developed.

As a tool for studying MPLIS development within this altered perspective, we have proposed a model that describes the full-life-cycle of community-wide systems over time. It is based on a review of previous research on the nature of GIS/LIS development in organizations, the results of surveys of GIS/LIS implementation activity, and anecdotal and personal experiences of the authors. The model describes the necessary stages in a linear development process in a community as well as indicators of those stages, factors that determine a change from one state to another and benefits. The model is intended as a common basis for understanding of and communication about

system development by decision-makers, system developers, and academic theoreticians equally well.

In addition to the theoretical model, we provide survey results showing that local governments are beginning to achieve democratization and suggesting that more communities need to focus on reaching that final stage of development. We also provide anecdotal evidence showing ways that communities are already accruing equity.

MPLIS DEVELOPMENT MODEL

We have proposed a theoretical model of multipurpose land information systems development with a community-wide perspective (Tulloch *et al.* 1996). The MPLIS development model includes several elements which make it unique, including factors that determine change, indicators of status, benefits, and stages of MPLIS development. The model also makes strides towards satisfying a series of criteria which have been lacking in previous models. The model can only be described briefly in this paper but has been presented previously in detail (Tulloch *et al.* 1996).

Reasons for proposed model of MPLIS development

The model of MPLIS development seeks to establish a general description of the complex manner in which these systems develop in a community. While previous research provided an important foundation, there remained some specific questions about system development that were unanswered. One of the primary questions that is unresolved is that of the appropriate perspective for model building. Retrospective case studies and studies of the adoption and diffusion of technology typically emphasize the perspective of someone within the organization or agency (or a few closely-connected organizations or agencies) that introduce the system and is limited in its scope to that organization. The apparent nature and extent of system development at this time in many agencies at all levels of government strongly suggests that a perspective on system development that encompasses a larger segment of the community is appropriate and needed if the system status and direction of change are to be understood and used to the benefit of both the organization and community. Aside from Crain and MacDonald (1984) there was little recognition that MPLIS development in a community continues beyond some initial point of "implementation" or "operation" within an organization to be fully developed or successful. Today, we need a model that recognizes that the full life-cycle of system development includes non-technical development such as increased participation in land-related decisions based upon access to and use of geographic and land information by a broad constituency in a community and not simply the pre-operational and operational stages of development in an organization.

Among the other concerns were a lack of a common descriptive language for discussing this process, little recognition of the distinction between existing states of the system and the forces for change between states, a distinction that

requires a reliable and valid tool that identifies and measures the elements of system development, a need for clearer delineation of system benefits, and, finally, a model that serves as a stimulus to the intellectual pursuit of knowledge. In addition to these specific concerns, there was a separate broad criterion that must be satisfied by any model among a set of models before one emerges as accepted. The successful model, like all theories, needs to be tested, embraced, and used by the community of users. Even the best designed model is of limited value if it is not embraced and used appropriately.

As mentioned previously, much of the existing literature focuses on a system within a single organization, or within a few loosely connected organizations that establish an enterprise-wide system. Our model is intended to apply to community-wide multipurpose systems, within the concepts of a local government MPLIS as set forth by Brown and Moyer (1989). This implies that the system is being developed for a community of users and beneficiaries, not simply to enhance the performance of an agency.

Elements of MPLIS development

The model was built largely upon the existing literature and related research performed by the authors. It consists of the following major elements; stages, factors, indicators, and benefits. The model began with a review and consolidation of implementation models that appear in the literature (e.g. Rogers 1962; Crain and MacDonald 1984; Vastag, Thum, and Niemann 1994) to produce a simple six-stage model. In this model, stages are identifiable and measurable states of MPLIS Development. The six stages, in order of progression, are: no modernization, system initiation, database development, recordkeeping, analysis, and democratization. Its is recognized that stages sometimes overlap but generally occur in the indicated order.

Indicators are the measurable variables that define the identifiable and measurable stages of MPLIS development. The model relies upon seven categories of indicators, further divided into fifty-three specific factors. For example, the nature and extent of data in a digital format can be used as an indicator of a system that has entered into the recordkeeping stage.

Factors emerge from both the practical professional literature (e.g. Croswell 1989) and the academic literature (e.g. Onsrud and Pinto 1993). Factors are the measurable social, economic, legal, institutional, technological, political, and cultural variables that determine the forces for change from one stage to another. The model relies upon twelve categories of factors, further divided into seventy-six specific factors. Funding is an example of a factor whose nature and extent can promote progress to more advance stages, prohibit forward progress, or even cause a system to retreat to previous stages.

The final element, benefits, are broadly defined as the identifiable and measurable components of the existing or anticipated community well-being achieved through MPLIS Development. It is not simply the well-being of a particular organization. Some specific benefits can also serve as specific

indicators of a particular stage when the existence of those benefits are common and specific to that stage. Benefits to the community are composed of three categories: efficiency, effectiveness, and equity.

The model of MPLIS development also further describes the relationship between these various elements. For this paper, we will simply point out a strong association between the final three stages of development and the three benefits. Efficiency is generally identified with recordkeeping, effectiveness with analysis, and equity with democratization.

BENEFITS: THE THREE E'S

How to describe the benefits obtained from system investments remains a significant issue. Economists recognize that benefits are both tangible and intangible. The identification of tangible benefits and their measurement in monetary terms make it possible to calculate a benefit/cost ratio for investments. However, economists also emphasize that it is equally important to identify and characterize intangible benefits and introduce these in any discussion of potential system investments. The model described in this paper is designed to incorporate attention to these aspects.

The model places the benefits of MPLIS development: into three broad categories: efficiency, effectiveness, and equity. Efficiency results where traditional activities are performed at a reduced cost, generate more products, are accomplished more quickly or in some combination. Effectiveness results when more or better information is generated from traditionally available data because of digitally stored data and the software for sophisticated analysis of that data. The emphasis shifts from benefits associated with traditional tasks to benefits associated with actions that rely upon system products. Equity results from a perceived or real increase in effective participation by citizens and organizations in decisions about land and resources. Although too space consuming to include here, the model also provides a mathematical expression of a system's total benefits (Tulloch *et al.* 1996)

Traditionally, attention to system benefits has focused on efficiency and effectiveness (Smith and Tomlinson 1992; Antenucci 1991; Gillespie 1994). Efficiency is an especially popular measure of benefits because is it easily expressed in monetary terms reflecting savings through system implementation. Effectiveness is somewhat more difficult to determine. However, estimates of monetary values of these benefits can be made based upon the savings associated with activities permitted by additional information generated from the old data. These are also popular measures of benefits because they relate closely to the internal needs of organizations and are easily understood by non-technicians and decision-makers in those organizations.

Recently attention has turned to the benefits associated with the broader use of system products beyond the organization and throughout the community. These related outcomes have been characterized as societal

benefits (Clapp *et al.* 1989), equity (Kishor *et al.* 1990; Cowen 1994), decision making (Pinto and Onsrud 1995), and democratization (Mead 1994; Lang 1995). As a general trend, this attention is evidenced by the amount of research concerning community issues such as public access (e.g. Epstein and Roitman 1987; Epstein, Hunter, and Agumya 1996) and concerns about the outcomes associated with the growing relationships between GIS and society (e.g. Pickles 1995; Sheppard 1995; McMaster *et al.* 1996; MSC/NRC 1997). Here the set of product users is potentially much greater in number and type than those users in the organizations that initiate and develop the system. The perspective encompasses the whole community of public and private organizations and people interested in the value to them in their land-related work that comes from the products of MPLIS Development. The context for these benefits is the full array of public and private policy plans, decisions, and actions where use of land and its resources are allocated.

The benefits from the use of system products by increased numbers of community members are labeled equity benefits. This label is appropriate because land-related decisions made by a more representative segment of the community means that GIS/LIS becomes democratized. This democratization represents an important advance in MPLIS development, holding out the potential for full community awareness, utilization and support for the system based upon a sense of increased and balanced participation in the allocation of land and its resources. Emphasis on these types of benefits is another important and different aspect of the community-wide perspective that informs the model.

There is also another important reason for describing these benefits as equity benefits. Its achievement depends upon wide distribution and easy access to system products. This fundamental characteristic of dissemination and access has always been a standard for public data and information and a subject of policy controversy (e.g. Epstein 1991; Brown 1992). The long-term democratic interest in the dissemination of records and information used by governments to execute their legislative and legal mandates is exhibited by the state and federal open records and freedom of information laws which are built upon the democratic principal that access to material used by governments for their public business is essential in a society where people need to know what their governments are doing. A negative view of broad, easy access by many to GIS/LIS data restricts the full development of systems. Access to information is the key to garnering the large unrealized potential for equity benefits.

RELATED FINDINGS

Survey Results

Recently, local governments in three states have been surveyed as part of a study of MPLIS development. Details of the survey methods and results have been provided elsewhere (Tulloch *et al.* 1996). However, a few specific results are directly relevant to this discussion of equity. The first of these is simply the perspective of respondents on the status of system development in their

local government. Approximately 5% of the respondents in both Wisconsin and Ohio indicated that their local government had achieved some degree of democratization (Figure 1). While any number of issues might be used to cloud the specific meaning of this response (e.g. respondents ability to understand the definition provided in the survey questionnaire), it still seems clear that a number of land information professionals recognized equity as an outcome of their system.

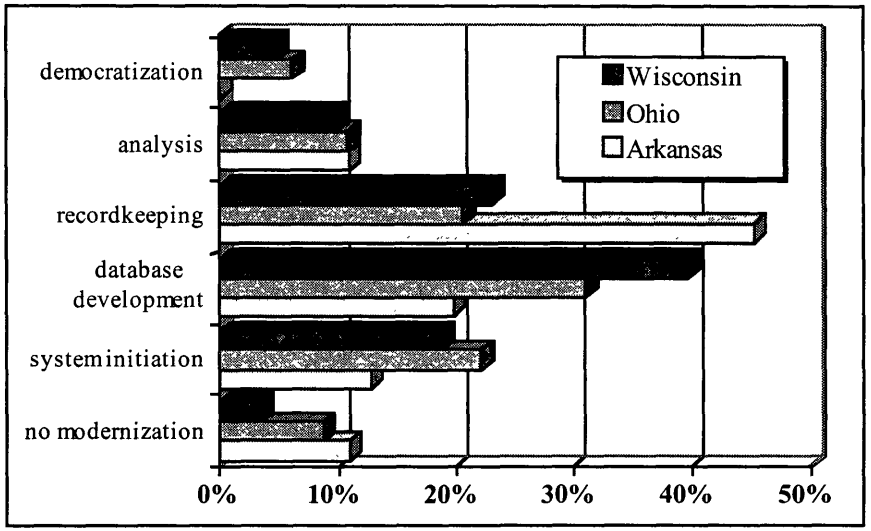


Figure 1 – The Stages of MPLIS development having been achieved by local governments in three states

Another question asked about the categories of indicators which might exist as a result of MPLIS development. Wisconsin respondents indicated that "use of technology" and "transfer of data" were the most commonly present indicators. In contrast, they also responded that "change in decision making processes" and "impacts on land related decisions" were the least present indicators, although they were shown as being present in some organizations. This seems to suggest, like the stages results, that most systems are not yet sufficiently developed to produce equity. It does seem to show the possibility of an increase in future societal benefits.

Anecdotal evidence

Because system benefits are usually accrued only after several stages of the MPLIS development process, it can be difficult to find examples of benefits, especially equity, as an outcome of system development in communities. However, we will endeavor to present a few recent occurrences which demonstrate the potential of this benefit.

When the land information officer in Waukesha County, Wisconsin automated the tax assessment records, a few "lost" parcels were discovered. These parcels accounted for thousands of dollars in lost revenue each year. The

first benefit generated was that of a more efficient operating office -- faster and cheaper to maintain. The second benefit came from the office's use of the system to "better" perform their traditional tasks, as illustrated by the discovery of the "lost" parcels.

For roughly a decade, Winnebago County, Wisconsin has been developing a MPLIS to serve its 140,000 residents in as many ways as possible. In recent years the database development has been completed and the system has been maintained and used for a variety recordkeeping and analysis purposes. One unexpected opportunity arose when investments in system development led to the discovery of inaccuracies in the FEMA produced Flood Insurance Rate Maps (FIRM) of the county. The maps are adopted by localities and used to determine whether homeowners are required to purchase flood insurance and also to determine where new structures can and cannot be built. In response to concerns raised about the maps, the county used existing data to construct new floodplain models and maps. The county system showed that, in fact, a number of homes had been inappropriately located in or out of various zones. Because of the high quality of the data and the "transparency" of the analysis, the county's efforts have been recognized as a suitable alternative to the old maps.

System efficiencies allowed for the timely production and update of necessary maps, address lists, and other system products. The result of the analysis -- a complex hydrologic modeling process not previously feasible – was a more effective means by which the county could fulfill its mandates. Both of these examples also have the subtle potential for the social benefit of equity because they may have both altered the public's confidence in those government agencies and may have a long-term impact on participation.

The Register of Deeds in Dane County, Wisconsin, has automated the title records and indexes for the county and maintains the digital material as the county's official property records. The entire database is available in digital form at the cost of reproduction (as mandated by state open records laws). Title insurance companies, for whom a complete copy of all relevant land transactions is necessary for competitive operation, are able to cheaply acquire the entire database for their use. While the existing firms are able to use this database to update their existing data at low cost, thereby increasing profits, new firms are able to enter the market avoiding the previously prohibitive start-up costs associated with building a database. The result has been the entry of new title insurance companies and an increase in competition which appears to have driven the cost of title insurance for homebuyers down about $300. This system, with an estimated cost of $500,000, results in a savings for homeowners of an estimated $6-7 million annually.

The Register of Deeds Office finds that the system makes its basic recordkeeping activity more efficient, requiring less time and space. Effectiveness is harder to judge because the primary duties of the office are recordkeeping and promoting access to data rather than an application that requires sophisticated data analysis. The automated system allows the general

public to access land records easier and faster with an increase in direct citizen participation in use of the material and an indirect increase in participation through the added title insurers. Hence the accrual of the equity benefit. The reduction in title insurance illustrates and provides measures for the benefits of this increased participation.

SO WHAT?

A fundamental lesson learned from the model's development and application concerns the relationship between time and equity. The results suggest that only after MPLIS development is well advanced does a system tend to produce its single most valuable benefit, equity, in the form of increased participation in land decisions with the resulting sense of greater fairness that characterizes a democratic society. Since recent measures of systems status indicate that few systems have yet reached democratization, it seems likely that the major societal benefits have yet to be realized by communities. Experience with many systems indicates, however, that an increasing number of communities should soon be able to capture this benefit at increasing levels.

For those working with communities to develop MPLIS, the model and its application provide a means to demonstrate and measure the outcomes of the development process. Moreover, the model shows the importance of continuing MPLIS development beyond "operationalization". Only through complete systems development can a community reap these most important benefits.

A community-wide perspective is becoming increasingly important. Potential beneficiaries are often people without technical knowledge who will be dependent, implicitly or explicitly, on the products and services of systems for their land-related activities. As they become aware of these systems, they will be in a position to influence the establishment of mandates and standards for MPLIS Development in the community. They, as representatives of the larger community, are likely to play an increasing role in determining the nature and extent of system development.

MPLIS Development is at a point of transition from system development initiatives that arise from the efforts of technically-oriented system builders to initiatives based upon the demands of many outside the organizations who want and need products for their decisions about land and resources. These potential users are greater in number, financial strength, and societal impact than the typical GIS technician in a traditional organization. These users, some of whom want data and products so that they can better participate in land and resource decisions in a community, represent groups that have yet to benefit in great numbers or to great extent from system development. However, many are increasingly aware of what the products of GIS/LIS can provide for them and are now prepared to exert influence on MPLIS development from outside the domain of traditional GIS/LIS organizations. Their interest, involvement, and impact is now crucial to the pace of development.

Finally, the model suggests that systems that are not fully developed (i.e. do not achieve the level of participation that represents democratization) are also most likely to eventually fail. It seems apparent that communities who recognize, demand, and plan for equity will be those communities whose systems are most likely to experience continued success and community support. We would go so far as to assert that, if these technologies are to experience continued success throughout the coming decade, a clear recognition of equity as a key concept in MPLIS development must emerge.

While efficiency and effectiveness alone can justify the cost of a system, the benefit of equity has the potential to be the largest of all when received by the community. In the long term, failure to secure the democratization associated with increased participation exposes systems to threats of cuts and elimination. A full realization of equity can lead to community-wide acceptance and even embrace of a system thus providing support for system activities.

Equity has many emerging, embedded concepts. A significant element is more extensive involvement by organizations and citizens in land related decisions. Indeed anecdotal evidence of minority groups being empowered by this technology is already emerging (e.g. Native American populations in Wisconsin). However, equity also includes public access to data, increased participation in government processes, altered community resource-related decisions, increased confidence in government and reductions in home owner land records processing costs such as title insurance. This trend, especially when viewed within a community-wide context, suggests not only the potential but the absolute need for community application of systems in order to see their use continued. As recognized by the FGDC's NSDI initiatives, this also suggests that local governments are becoming increasingly vital players in the production, use, and dissemination of land information.

BIBLIOGRAPHY

Antenucci, J. C., K. Brown, P. L. Croswell, M. J. Kevany, and H. Archer. (1991). *Geographic Information Systems : A Guide To The Technology.* New York : Van Nostrand Reinhold.

Brown, K. (1992). A Response to Earl Epstein. URISA Journal, 4: 6-8.

Brown, P. M., and D. D. Moyer (eds.). (1989). *Multipurpose Land Information System: The Guidebook*, Washington, DC: The Federal Geodetic Control Committee.

Clapp, J. L., J. D. McLaughlin, J. G. Sullivan, and A. Vonderohe. (1989). "Toward a Method for the Evaluation of Multipurpose Land Information Systems," *URISA Journal,* 1, 39-45.

Cowen, D. J. (1994). "The Importance of GIS for the Average Person," in *GIS in Government: The Federal Perspective*, Proceedings of the First Federal Geographic Technology Conference, Washington DC, 7-11.

Crain, I. K., and C. L. MacDonald. (1984). "From Land Inventory to Land Management." *Cartographica*, 21, 40-46.

Croswell, P. L. (1989). "Facing Reality in GIS Implementation: Lessons Learned and Obstacles to be Overcome," in *URISA '89 Conference Proceedings*, 4, 15-35.

Epstein, E. F. (1991). In My Opinion. In: URISA Journal, 3 (1): 2-4.

Epstein, E. F., and H. Roitman. (1987). "Liability for Information," *URISA '87 Conference Proceedings*, Vol. 4, 115-125.

Epstein, E. F., G. Hunter, and A. Agumya. (1996). "Liability and Insurance for the Use of Geographic Information." *URISA '96 Conference Proceedings, Vol. 1, 294-301.*

Gillespie, S. R. (1994). "Measuring The Benefits of GIS Use: Two Transportation Case Studies." *URISA Journal,* 6 (Fall): 2, 62-67.

Kishor, P., B. J. Niemann, D. D. Moyer, S. J. Ventura, R. W. Martin, and P. G. Thum. (1990). "Lessons from CONSOIL Evaluating GIS/LIS." *Wisconsin Land Information Newsletter* 6:1, 1-11.

Lang, L. (1995). "The Democratization of GIS." *GIS World*, 8:4 , 62-67.

Mapping Science Committee, National Research Council. (1997) *The Future of Spatial Data and Society: Summary of a Workshop.* Forthcoming. National Academy Press: Washington, D.C.

McMaster, R. B., B. J. Niemann, Jr., S. J. Ventura, D. D. Moyer, D. L. Tulloch, E. F. Epstein, and G. Elmes. (1996). "GIS and Society," University Consortium for Geographic Information Science White Paper.

Mead, R. A. (1994). "Field-Level Diffusion Eases GIS Implementation Efforts." *GIS World*, 7:11 (November), 50-52.

Onsrud, H. J., and J. K. Pinto. (1993). "Evaluating Correlates of GIS Adoption Success and the Decision Process of GIS Acquisition." *URISA Journal,* 5:1, 18-39.

Pickles, J. (ed.). (1995). *Ground Truth: The Social Implications of Geographic Information Systems*, New York: The Guilford Press.

Pinto, J. K., and H. J. Onsrud. (1995). "Sharing Geographic Information Across Organizational Boundaries: A Research Framework," in *URISA '95 Conference Proceedings,* Vol. I: 688-694.

Rogers, E. M. (1962). *The Diffusion of Innovations,* New York: Free Press of Glencoe.

Sheppard, E. (1995). "Geographic Information Systems and Society: Towards a Research Agenda." *Cartography and Geographic Information Systems*, 22, 5-16.

Smith, D. A., & R. F. Tomlinson. (1992). "Assessing the Costs and Benefits of Geographical Information Systems: Methodological and Implementation Issues." *International Journal of Geographic Information Systems*. 6: 3, 247-56.

Tulloch, D. L., E. F. Epstein, and B. J. Niemann, Jr. (1996). "A Model of MPLIS Development in Communities: Forces, Factors, Stages, Indicators, and Benefits," *GIS/LIS '96 Proceedings* 325-348.

Vastag, P. H., P. G. Thum, B. J. Niemann. (1994). "Project LOCALIS: Implementing LIS/GIS in Local Government." *URISA Journal*, 6:2, 78-83.

ISBN 1-57083-043-6 (set)
ISBN 1-57083-048-7