HIERARCHICAL GRID STRUCTURES FOR STATIC GEOGRAPHIC DATA BASES

Andreas Kleiner and Kurt E. Brassel
Department of Geography
University of Zurich
Winterthurerstr. 190
CH-8057 Zurich, Switzerland

Abstract

Projects for the creation of large cartographic data bases and the advent of high volume optical disk technology create a need for structures for static mass data storage. This paper reviews existing approaches for structuring mass storage and proposes a scheme which combines the following concepts: a) separation of "background data" from "geographic object data"; b) use of hierarchical grid structures to define data buckets; c) coordinate shuffling to define keys for the spatial domain; d) use of efficient one-dimensional access structures (B-trees, Extendible Hashing); and e) pointer elimination due to the static properties of the data bases.

1. Introduction

Efforts are undertaken on both international and national levels to systematically collect base map information for digital data processing. The information contents of large area base map series are very high and the resulting data files are extremely voluminous. The use of adequate data base structures is thus a most important requirement for efficient utilization of large data bases. In recent years a multitude of data base models have been proposed. The present paper discusses some alternative storage approaches for data handling based on bulk memory spatial data bases. It is evident that storage efficiency cannot be defined in absolute terms, but it is dependent on the particular usage characteristics of the data. An example of such a large data base is the creation of a digital version of a worldwide 1:1 million map series, including hydrology, administrative boundaries, transportation features and relief (Bickmore 1986). It will be basically a worldwide multi-layered base for environmental sciences consisting of a mixture of point, line, area and volume types of geographic objects. As a product it is envisioned to be offered on one or multiple compact disks as a read-only store for easy distribution. Based on such a scenario we discuss some alternative storage structures for static base map elements of various characteristics. Dynamic map modification and updating is of no concern here; our aim is to provide a structure for fast search and retrieval of background elements. Though this information will be combined with a variety of "foreground" elements of the specific users, we shall disregard the aspects of foreground elements.
2. Existing Approaches for Structuring Mass Storage

Recent efforts of data handling in computer cartography were mainly concerned with data structures which did not need much space on external storage devices and were concentrated on efficient internal processing. Among them quadtree structures are of special interest; they appeared in the 1970s (see a review article by Samet 1984). Our discussion will also include the bintree or 2-d-trie (Knowlton 1980, Tamminen 1984, Orenstein 1982) which uses a binary rather than a quaternary tree structure, that is, each father node can have only two son nodes with alternate splits in each spatial direction. Two levels of a bintree correspond to one level of a quadtree.

Large data bases have to be stored on external storage media. Traditionally, bucket methods are used for this task. In the case of multidimensional data dealing with the geometric organization of space, bucket methods are also called cell methods (Tamminen 1981b). These are based on the principle of storing spatially adjacent objects in adjacent physical stores and pack the data into blocks or pages just large enough to read them into central memory by a single disk access. Various strategies exist to subdivide space into hierarchical buckets of cellular or grid shape.

Bucket methods can be grouped into two basic approaches (Nievergelt et al. 1984). One group of methods structures the data by trees of keys and searches on the basis of guiding discriminators ("signposts"), while the others partition the underlying (coordinate) space. The latter allow access to a directory or directly to the data by address computation. Some structures originally developed for internal processing may also be used for the management of data buckets. Initial research dealt with the access to one-dimensional data. The structure used most often is the B-tree, which is a generalization of the binary search tree as adapted for external block access operations (Bayer and McCreight 1972). A special variant suitable for range searches is the B+-tree (Comer 1979). Other schemes are Extendible Hashing (Fagin et al. 1979) which stores a bucket pointer for each potential hash value, and Linear Hashing (Litwin 1980) which allows hashing without a directory. Since multidimensional search problems are of special interest for the geo-sciences, efforts to develop multidimensional access methods were undertaken by both computer scientists and specialists in spatial data handling. Examples of structures include extensions of the B-tree such as the multidimensional B-tree (Scheuermann and Ouksel 1982) or the k-d-B-tree (Robinson 1981) as a combination of the B-tree and the k-d-tree (the multidimensional binary search tree; Bentley 1975). Proposed methods which organize data space are Multipaging (Merrett 1978, using directory scales), Dynamic Multipaging (Merrett and Otoo 1982), EXCELL (Tamminen 1981b; Extendible Hashing for multi-dimensional space), the Grid File (Nievergelt et al. 1984; similar to EXCELL, but with internal linear scales and an external directory) and methods of Linear Hashing (Ouksel and Scheuermann 1983, Burkhard 1983, Orenstein 1983). The Field Tree (Frank 1983b) as a quadtree structure is based on a regular partitioning of space and is accessed by tree traversal. Most of these structures are designed to manage dynamic data bases.

Some authors deal with the problem of storing spatial objects without splitting them
along cell borders. Most methods, however, split the objects and have to concatenate them each time they are extracted. Hinrichs (1985) proposed a method of storing objects through parameters (center point and half side-length of the enclosing box) in higher dimensional space. However, access becomes extensive due to conical search areas. Frank (1983b) and Abel and Smith (1983) used a quadtree organization of space to store objects in the smallest enclosing quadtree cells. Frank (1983a) introduced a derivation of the quadtree where the cell origins are systematically shifted for the respective hierarchical levels; this allows all objects to fit a bucket cell of appropriate size and avoids that small objects at high level quadrant borders have to be placed in high level cells due to the border coincidence. To our knowledge Frank did not follow up on this method due to the complexity of access processes (Frank 1983b).

Another important concept for multidimensional data handling is coordinate "shuffling", the operation of bitwise interleaving coordinate keys, resulting in so called Peano keys (Peano 1973). It was developed in the domain of GIS for the transformation of multidimensional space into one dimension and results in "linear quadtrees" (or linear bintrees; see Peuquet 1984, Samet 1984). Access on these keys may be provided by methods for one-dimensional data (Tamminen 1981a, Abel and Smith 1983). There has also been a growing interest in this concept in the domain of conventional data base research (Tropf and Herzog 1981, Ouksel and Scheuermann 1983, Burkhard 1983, Orenstein 1984).

3. Hierarchical Grid Structures for Static Geographic Data Bases

The purpose at hand is to design storage strategies for spatial data on read-only external mass storage devices. A first requirement is to minimize both access time and external storage space. Under the assumption that information used for mapping or spatial analysis is highly clustered, bucket methods are typically used for this task.

Another major issue is the definition of the objects to be stored in the grid cell storage buckets. Spatial objects by definition may be of point, line, area or volume type. The latter three may be arbitrarily cut into parts by cellular grid systems. Depending on the application this may or may not be acceptable. For the purpose of this discussion we distinguish between those objects which can be subdivided along grid borders and those that should be stored in their integrity within one single data bucket. We call the former "background data", the latter "geographic object data". Background data can be stored and retrieved more efficiently as long as objects are not to be reconstructed.

In our task of storing large volume spatial data on read-only memory we propose to combine the following concepts: a) separation of "background data" from "geographic object data" and use of specific spatial bucket or cell methods for both cases; b) coordinate shuffling in the spatial domain; c) use of efficient one-dimensional access structures; and d) taking advantage of the static properties of the data to reduce storage overhead and processing time.
In our presentation we shall proceed as follows: First we present the basic concepts and specific schemes for the background data (3.1), then explain the specific organization of the geographic object data (3.2), and close with a discussion of some alternatives for the organization of keys and adaptations for static data bases (3.3).

### 3.1 Hierarchical Bucket Methods for Storing Background Information

Background data include point features and elements of linear and area objects that are separated at grid cell borders. A reconstruction of object integrity and topology is not required. Information pertaining to adjacent objects should be stored whenever possible in one data bucket. For internal storage quadtree or bintree structures appear to be optimal. On the other hand these structures are not suited for external storage of the tree nodes because tree traversal is inefficient. Analog to the development of the B-tree from the binary search tree where the number of sons of each father is so big as to occupy one page, we could extend a quadtree to a hierarchical grid structure with a whole matrix of son cells for each father cell. This minimizes the number of disk accesses and reduces search time not only by physical clustering of adjacent data, but also by minimizing reading operations in the access structure. This concept is illustrated in Figure 1. While solving one problem we diminish spatial flexibility, reduce cell occupancy and thus processing efficiency. To alleviate these problems we separate the organization in the space domain from the organization of bucket access: Space is subdivided into hierarchical cells (e.g., quadtrees, bintrees) which are then managed by shuffled keys and one-dimensional access structures (e.g., B-trees). The specific cell units represent data buckets containing background elements in form of point and string data. The size of the cells is a function of the information density in space. The model used most frequently is the quadtree. However, in cases where other than raster type data are to be handled bintrees are even more flexible. If a bucket is overfilled, it is first split into two sons instead of four. Figure 2 shows the bintree for a hypothetical map. All cells are labeled by shuffled codes interleaving bits in x and y directions. The number of digits used implicitly indicates the cell size or hierarchical level. The shuffled codes of the data buckets are stored in B-trees (or other access structures) where the B-tree elements again are packed in pages or "index-buckets" (Figure 3a).

### 3.2 Storing Geographic Object Data

For the storage of spatially extended objects which have to be handled as integral entities we will use an adaptation of the cell organization proposed by Frank (1983a). In order to find a cell into which an object of arbitrary form and size can fit, cells of all levels of the hierarchy may be used in the same structure. A regular quadtree is chosen as proposed by Abel and Smith (1983), but with a displacement of the grid at each level. Small objects crossing high level borders now fit lower level cells, thus avoiding to force too many objects into the top cells. Instead of an implementation of rather complicated "trees" where sons can belong to two or four fathers (i.e., graphs), we propose to use bit interleaving of coordinates. As point of reference we
either choose the lower left corner or the center of each cell with the addition of a suffix specifying the level and implicitly the displacement with respect to a regular matrix. In the former case each cell has a unique key which avoids the need for a level suffix. This scheme is illustrated in Figure 4. The area of interest is bound by heavy lines. In the example this is a single cell of Level 1; it is overlaid by cells of Levels 2 and 3. Objects that fit into cells of Level 3 are addressed by Level 3 keys, objects that cross Level 3 borders in cells of Levels 2 or 1. Cells at all levels include pointers to a data store where the sequence order is defined by the shuffled location keys of the lower left corner of the cells. Not all values of the domain of the keys are actually used. Each father has 9 sons, of which 4 sons are shared with one neighbor and 4 sons with three neighbors. Range searches are performed by the sequential traversal of the tree where each father is visited after the first three sons and before the remaining six sons.

The list of shuffled codes again is sorted and packed into B-tree pages. Each shuffled code is associated by a pointer to a data bucket. Each bucket may include a number of objects of various types: point, line area or volume objects with or without topological identifiers. For a polygonal network data buckets may include both arc
I

Figure 2: Bintree organization for background data (empty cells are shaded)

(1-cell) and polygon (2-cell) objects. Arc objects may be defined by a sequence of points and pointers to adjacent arcs and polygons, polygon objects by strings of arc identifiers, etc.

In general, a polygon description may not happen to be stored in the same bucket as its arcs. It may, therefore, be reasonable to define a unique object identifier for all objects of any type. This identifier consists of the bucket code plus a sequential counter within each bucket. Such a scheme enables searches based on objects independent of spatial criteria.

3.3 Alternative Key Organizations

In the previous discussion we have proposed B-trees as an index structure to the data buckets. Due to the static nature of the data base, the presorted buckets are stored at fixed physical locations on a mass storage device. Since the data are sorted by shuffled keys, this results in the important advantage that sequential searches in space
Figure 3: Access structures for the bintree in Figure 2: a) B-tree b) EXCELL
can be executed by sequential reading from disk. We therefore do not need a B+-tree for range search as proposed by Abel (1984). Also, if the structure of the B-tree is exhaustively defined, all pointers to B-tree pages and data buckets can be eliminated since their external storage position can be calculated from the position of the cell in the B-tree. Analog to "linear quadtrees" we call this structure a "linear B-tree".

Under certain circumstances the use of Extendible Hashing (EXCELL, Tamminen 1981a) may be a valid alternative. In contrast to the B-tree all area keys of smallest cell size are members of the directory. For higher level cells all pointers of its constituting subcells point to the same bucket. Area cells of any size can thus be addressed directly. For each area unit a pointer to the respective bucket location on mass storage is maintained. Figure 3b visualizes this alternative. The Extendible
Hashing option may be inferior to the B-tree option with respect to storage requirements but superior for random data access time. As we have pointed out above, not all values in the domain of keys are actually occupied in the case of geographic object storage. Since Extendible Hashing, however, uses entries for all potential keys, storage is inefficient. It is therefore advisable to compress the shuffled keys by a compression function. For each cell the compressed keys then indicate the position of the pointer to the bucket in mass storage.

4. Conclusions

We have discussed various storage structures for read-only spatial bulk data. We recommend to separate background data from geographic object data. Background data (points and spaghetti lines cut at cell borders) are preferably stored in spatial buckets defined by bintrees which are labeled by shuffled area codes and organized in a B-tree structure. Due to the static nature of our task, pointers to buckets are not required. A valid alternative is the use of Extendible Hashing (EXCELL). Geographic object data are packed in hierarchically displaced structures of quadtree cells, which are identified by shuffled area codes. The sorted codes are organized either for B-tree or Extendible Hashing access. In the latter case compression of shuffled keys is advisable. In a static environment all proposed structures allow for sequential retrieval of data buckets without the use of an index structure. Final decisions on the type of structures depend on the specifics of the map data to be recorded and the circumstances of their use. A major decision will be as to which elements will be stored as background or geographic object data. Other parameters are the level of detail to be recorded, i.e., the total volume of background or object data. Given this quantity for both types of stores, a decision has to be taken with respect to the cell resolution for background and object stores; this decision is related to the size of the data buckets used.

Future efforts shall be devoted to the determination of the specific parameters of a real-world project and the analysis of its specific usage profile. Additional work relates to the creation of algorithms for the construction of the static data base and to system implementation and testing.

Acknowledgements

This project is supported by a contract of the Swiss government. Mr. A. Herzog and Dr. H. Kishimoto have kindly reviewed the text. These contributions are gratefully acknowledged.
References


DIGITAL DEFINITIONS OF SCALE-DEPENDENT LINE STRUCTURE

Barbara Pfeil Buttenfield
Department of Geography
University of Wisconsin, Madison WI 53706 USA

Abstract One problem encountered in developing an automated system for cartographic line generalization involves the procedures to modify tolerance criteria. Tolerance values should be modified at locations where the line structure changes significantly. This is not a simple determination, because the geometry of a cartographic line varies with the scale of its graphic depiction. Automatic modification requires numeric descriptions of scale-dependent geometry, and a consistent procedure for determining structural change at a variety of scales. The paper reports on implementation of the structural descriptions for a sample of cartographic lines, and proposes an application to generalization tasks.

INTRODUCTION

Attempts to measure and generalize the information contained in a cartographic line have developed along a range of approaches, from purely descriptive to nearly inferential. That these are more readily accomplished in a manual than a digital context is primarily due to the problems involved in expressing consistent guidelines for an essentially intuitive process, which is the retention of recognizable characteristics of the line during its generalization. The geometric and visual cues used to recognize a particular geographic feature are scale-related, and the sheer volume required to digitally encode all possible cues at all possible map scales is prohibitive.

This paper suggests an alternative, procedural approach for automating descriptions of cartographic line structure, which distinguishes between geometric characteristics in terms of graphical rather than perceptual aspects. The advantage is to permit digital definitions of structures which vary depending on the scale of their graphic representation. The intention is to focus on generalization as a process of construction as much as recognition.

The metaphor used in the research is common to remote sensing. As a specific surface cover is characterized by a predictable spectral response over a range of wavelengths, spectral signatures may be used to distinguish between surface covers, and to identify them. Rather than a spectral progression over a range of wavelengths, this research measures a 'structure signature', a geometric progression over a range of graphic scale. A particular cartographic line can be expected to have a predictable progression over a range of map scales; and structure signatures can be used to distinguish line features.
Structure signatures have been developed for four sample cartographic lines, to demonstrate the metaphor (Figure 1). These lines were selected for a particular reason. Prior to the rise of automation, cartographers (see for example Raisz, 1948; Pannakoek, 1962) insisted that the key to retaining the character of a feature during generalization lay in a knowledge of its underlying geomorphology. Four distinct geomorphic processes are reflected in these lines. The FJORD line displays glacial downcutting and submersion; the emergent coastline called HUDBY shows the result of isostatic rebounding following glacial recession. TEXAS is a depositional feature. The dendritic pattern in SIOUX is formed by the passage of water across a homogeneous bedrock. Digital definitions for graphic structure should be as distinct as are the geomorphic descriptions.

![Figure 1. The four sample cartographic lines.](image)

**BUILDING THE STRUCTURE SIGNATURES**

The mechanics of deriving a structure signature is accomplished hierarchically, building up a set of geometric measurements through a sequence of resolutions. Following Poiker's (formerly Peucker, 1975) theory of bandwidth encoding, and the line reduction algorithm associated with it (Douglas and Peucker, 1973), one first measures the geometry of the entire line, then breaks the line into two pieces and measures the geometry of each piece, breaks up those pieces, and so forth.
With each subdivision, the coordinate endpoints for each segment are stored, along with the measured geometry, in a strip tree data structure (Ballard, 1981). The term 'strip' in this paper is used in a similar context to Poiker's use of the term 'band' in his bandwidth theory. Each row of the strip tree contains a representation of the entire line: the first row contains the line in one piece, the second row in two pieces, then in four, etc (Figure 2). Nodes in the tree (called strips) represent line segments. Traversing any row of the tree enables a comprehensive description of the geometry for a given level of resolution.

Figure 2. Building a strip tree of the subdivided line.

Two types of geometric measures are stored at each node of the data structure, along with the coordinate endpoints for the strip. The first type of measure is based on the bandwidth concept (Figure 3). Length of the strip is measured as the Euclidean distance between the endpoints. The straight line segment connecting these endpoints is called an anchor line. The width of the rectangle is determined by the maximum perpendicular deviations on either side of the anchor line. Also recorded is the location along the anchor line where the next subdivision takes place, and this measure is called segmentation. Measures of width and segmentation are standardized to the length of their anchor lines, to allow for subsequent comparisons between sample cartographic lines.
Band Length \[ L = \sqrt{(X_b - X_e)^2 + (Y_b - Y_e)^2} \]
Band Width \[ W = \frac{(A + B)}{L} \]
Maximum Deviation \[ M = \frac{A}{L} \]
Segmentation \[ S = \frac{C}{L} \]

Figure 3. Length-based measures of band width geometry.

Two other geometric measures have been recorded for each strip. First is the error variance, that is, the sum of the squared deviations of the original coordinates from the derived anchor line. Second is a measure of monotonicity, defined as the number of times the original coordinate string crosses its anchor line. By standardizing this measure to the number of coordinates in the original string, a probability measure is derived: given that point \( n \) is on a particular side of the anchor line, what is the probability that point \( n+1 \) will be on the same side? For a circular arc, the monotonicity value equals 0.00, and values approaching 0.99 the geometry of a coordinate string which crosses the anchor line between every pair of points. A value of 1.00 by this definition implies a coordinate string which lies directly upon the anchor line.

The structure signatures are built from these measures, by summarizing across rows of the strip tree. Computing a mean and standard deviation for the measures in each row provides a summary of the graphic structure at finer levels of resolution. The progression of geometric measures from one row of the tree to the next reflects the structural response across a range of graphic resolution. Structure signatures are stored in digital form, as a look-up table of mean and standard deviation values. This is the form enabling automatic decisions about when to modify tolerance values in generalization. However, it is probably easier to understand the structure signature concept by considering the measures in graphic form (Figure 4).
INTERPRETING THE STRUCTURE SIGNATURES

A brief discussion of the various signatures may serve to explain how it is that these scale-dependent descriptions may be used to distinguish between cartographic line features. In the computer, of course, these distinctions are made statistically, and this will be covered presently. Please note that, for all the plots in Figure 4, on the facing page, the numbers 1-5 on the x axis represent the five rows in the strip tree, corresponding to five iterations of subdivision applied to each line. Measures in each row can be summarized by mean or by variance, these plots display structure signatures for the mean.

In reading this discussion, one should keep in mind that these strip trees contain fewer rows than would probably be encountered in a realistic generalization task. Definitive interpretations of particular graphic structures are not realistically based on such small samples, of course, and no reliable conclusions about the specific graphic structure of a fjord or dendritic river pattern should be drawn. Rather, the discussion which follows is intended to explain how such signatures may be interpreted, and used to distinguish between different types of line features.

The first structure signatures display widths of the strips on both sides of the anchor line. Think of the x-axis of the signature as an anchor line; lying on each side of this anchor line is a summary of strip widths for the positive and negative sides of the anchor line. The degree to which strip widths are mirrored on either side of the anchor line should be considered as cross-sectional symmetry, measured at several levels of resolution. SIOUX appears to be most symmetric, while the two sides of TEXAS seem discrepant.

One may also compare the slopes of these plots, as the rate at which bandwidths decrease with resolution -- the varying slopes indicate clear differences in this decrease from one line to the next. One could interpret this measure as the rate at which large details are lost during generalization, as it summarizes the maximum deviations from anchor lines. As a geologist might examine a thin section of a rock or soil sample, these signatures are presented as thin sections of a cartographic line.

The second row of structure signatures display error variance measures, computed for each strip as the sum of squared deviations of original coordinates from the anchor line. As expected, error variance decreases with finer resolution as more strips are included to describe the line, more pairs of coordinate endpoints tie the strip tree representation back to the original coordinate string. As individual deviations of the generalized representation from the original line decrease, the sum of squares will drop.
In effect, error variance may be interpreted as a rate at which small details are lost during generalization. (This interpretation contrasts with the cross-sectional symmetry measure, as an index of the loss of large details along the line.) It provides an index of generalization error for the reduction algorithm which subdivided the lines; and as McMaster's (1983) research has already shown, one may expect that the index should differ for other generalization routines applied to this same sample of lines. Different generalization algorithms may produce representations of varying accuracy, but here is graphic indication that the quality of generalization for a single algorithm also varies between lines with scale and resolution.

In other words, what is interesting about these signatures is not that the decrease exists, but that the rate of decrease differs from one line to the next. The loss of details proceeds at similar rates for FJORD and TEXAS, indicating that the algorithm simplifies some types of features similarly. Three of the four signatures include a sharp elbow, which implies that this algorithm reduces details in a nonlinear progression for some but not all cartographic features. Verification of this implication will naturally require much larger coordinate samples, and should be evaluated for more than four types of line features. For the present, it is an intriguing question raised by the evidence of the structure signatures.

Interpretations for the segmentation structure signature reflect longitudinal symmetry for the lines. Segmentation occurs at the location of the maximum perpendicular deviation from the anchor line; and the measure is reported as fraction of total anchor line length. A value of 0.5, for example, indicates that the breakpoint lies exactly halfway along the anchor line. To the extent that a line is self-similar (Mandelbrot, 1982), the location of this deviation will remain in constant proportion at all levels of resolution, and the segmentation signature will vacillate about the value 0.5. In fact, this seems to be the case for SIOUX and HUDBY, for TEXAS the pattern is more pronounced. The final structure signatures summarize monotonicity, which is the number of times the original coordinate string crosses the anchor line. Monotonicity is not affected by the magnitude of details and deviations from the anchor line, but rather the frequency with which the original line oscillates about the anchor line. Rising monotonicity implies constant or increasing oscillation. At finer levels of resolution, one would expect the successively short coordinate strings to cross the anchor line less often, and to reach an equilibrium when the number of oscillations begins to drop. At this level of resolution, the structure signature will begin to level off.
What distinguishes one line from another here seems to be the level at which the equilibrium is reached. SIOUX reaches equilibrium early on, followed by TEXAS and then FJORD. The continuing increase of HUDBY through all five levels implies that the frequency of oscillations is preserved at finer resolutions, even though the magnitude of those oscillations (as measured by error variance) may not be changing at all.

The measures of strip width, error variance, segmentation, and monotonocity are summaries, mean values computed for rows of the strip tree data structure. Each measure summarizes a particular aspect (e.g., symmetry, oscillation) of the line geometry as it changes with graphic resolution. The structure signatures have been presented as visual demonstration that the geometry changes with changes in scale, and that the structural progressions differ between types of line features. It must be stressed again that the small sample does not warrant concluding that these signatures identify any particular geomorphic structure, but merely that they provide clear distinctions between graphic structures which are scale dependent. The next step involves statistical verification of these distinctions.

EVALUATING THE STRUCTURE SIGNATURES

If two lines have different structure signatures, one may conclude that their graphic structures are distinct within a range of resolutions. Stated another way, the geometric measurements plotted in the signature should provide significant discriminating power between lines of different graphic structure, within a range of resolutions.

To test this, the geometric measures were input to discriminant analysis. Five analyses were performed, at each of five levels of resolution. Preliminary correlations between the geometric measures indicated no systematic relations across the sampled range of resolutions. The purpose of the analysis was to ascertain the degree to which these measures may be relied upon to discriminate between types of cartographic lines. In theory, at the finest possible resolution, each strip will encompass only two points, monotonocity, error variance, and strip width will equal zero, and the structure signatures for all lines should be equivalent. Thus the discriminating power of should hold within a finite range of resolution.

Stepwise entry of variables has been utilized, to determine which parameters are the most powerful discriminators. Notice in the last column of Table 1 that at most levels of resolution, the two width parameters account for the largest amount of variance. It is interesting that after the first subdivision, neither monotonocity nor segmentation lend significantly to the analysis. The summed eigenvalues in the third column provide a relative measure of the total amount of variance in the set of discriminating variables.
Table 1. Statistical Summaries for Discriminant Functions.

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>FUNCTION</th>
<th>EIGEN VALUES</th>
<th>% VARIANCE EXPLAINED</th>
<th>SIGNIFICANT DISCRIM. COEFFICIENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1(.027)</td>
<td>3510</td>
<td>60.48</td>
<td>monotonicity</td>
</tr>
<tr>
<td></td>
<td>2(.050)</td>
<td>.2293</td>
<td>39.52</td>
<td>error var, monotonicity</td>
</tr>
<tr>
<td>2</td>
<td>1(.007)</td>
<td>.2555</td>
<td>61.44</td>
<td>negative width</td>
</tr>
<tr>
<td></td>
<td>2(.050)</td>
<td>.1076</td>
<td>25.86</td>
<td>positive width</td>
</tr>
<tr>
<td>3</td>
<td>1(.020)</td>
<td>.0953</td>
<td>64.59</td>
<td>positive width</td>
</tr>
<tr>
<td></td>
<td>2(.070)</td>
<td>.0522</td>
<td>35.41</td>
<td>*</td>
</tr>
<tr>
<td>4</td>
<td>1(.013)</td>
<td>.0768</td>
<td>44.40</td>
<td>error var, pos/neg width</td>
</tr>
<tr>
<td></td>
<td>2(.027)</td>
<td>.0587</td>
<td>33.91</td>
<td>error variance</td>
</tr>
<tr>
<td>5</td>
<td>1(.005)</td>
<td>.1019</td>
<td>79.15</td>
<td>neg/pos width, error var</td>
</tr>
<tr>
<td></td>
<td>2(.280)</td>
<td>.0181</td>
<td>14.05</td>
<td>*</td>
</tr>
</tbody>
</table>

* No coefficients are listed for non-significant functions.

NOTE: numbers in parentheses refer to significance levels.

Notice the sharp drop in the eigenvalue sum at Level 3, indicating a marked decrease in the discriminating ability of the parameters included at this point. In part this has to do with the fact that the parameters used to construct the functions are converging to zero, and becoming somewhat correlated; thus the discriminating power of the functions is reduced. As predicted at the beginning of this analysis, there is a limit to the meaningful distinction of lines by these measures, and this limit is dependent upon resolution.

A final part of the discriminant analysis utilized the discriminant functions to classify observations. For the present small sample of lines, the overall accuracy of classifications was not much better than chance; as anticipated, classification error increased steadily at finer levels of resolution, again indicating the limits to the range of the discriminant functions. The low accuracy of the classification suggests that the geometric measures by themselves are not sufficient to identify specific structures. However, the results of the discriminant analysis demonstrate that the geometric measures provide statistically significant distinctions between categories of cartographic line structures, that is, structures which exhibit scale-dependence. This discriminating power suggests an application of the structure signatures to the generalization problem discussed at the beginning of this paper, namely the automatic modification of tolerance values. This will be presented orally at the conference.
SUMMARY

This research has developed digital methods for description of cartographic line features, whose structure changes with the changing scale of graphic representation. It has been demonstrated that a hierarchic structuring of coordinate data facilitates derivation of the signatures, that a visual logic is evident in plotting signature values, and that the digital definitions may provide statistically significant discriminating capabilities between line features within a finite range of graphic resolution.

The structure signatures provide numeric information by which one scale-dependent structure may be distinguished from another. The signatures cannot by themselves provide sufficient information about line structure to identify a specific type of cartographic line, or to label it as a fjord, for example. The signatures are not intended to guide in the assignment of a specific tolerance value to a specific line. This will require a larger sample of line features, as well as perceptual verification of the visual logic map readers associate in generalization. However, the signatures may provide automatic capabilities to modify tolerance values, and this application is presented as a first step towards automatic line feature generalization.

LITERATURE CITED


Currently in the United States, more and more small government agencies are turning to automated systems for handling required land information. The benefits of automation can be great, and, with improved technology and increased competition among vendors, the costs have been drastically reduced. This paper describes the thought process in the development of a Land Information System (LIS) for a regional planning agency in the United States. The needs and goals of the users of the system dictated that it be flexible allowing both geographical data (maps) and attribute data to be stored and analyzed together. A dearth of microcomputer-based turnkey packages for this purpose in late 1984 forced the creation of a hybrid system that could combine a commercially available software package for attribute data (a relational database) with a software package for map analysis (Geographic Information System).

The Fifth Planning District Commission (PDC) is a regional planning agency of the commonwealth of Virginia located in the city of Roanoke, Virginia. The PDC jurisdiction includes four counties (1700 sq. miles), a Metropolitan Statistical Area of 250,000 people, and many small towns and cities (Figure 1). Land uses in the region range from urban-industrial to national forest. In January of 1983, the Fifth PDC convened a committee of representatives of constituent governments to determine those ways that the PDC could best assist in the orderly process of planning. By February, 1984, it was decided that the greatest single obstacle to effective planning efforts was the scarcity of readily accessible data. Costs for searching through courthouse records manually as each need arose were becoming prohibitive. A centralized, computerized data base was to be housed at the PDC for all to have access. Data could be required at many different scales, resolutions, and by various units of aggregation, therefore the land parcel was determined to be common denominator for all the expected needs. The system would consist of thirty-seven criteria
that would be collected for all land parcels in the four county area.

![The Fifth PDC Region](image)

**INITIAL PLANNING OF THE SYSTEM**

The greatest obstacle to system development was a rather limited budget. Many extensive systems have been developed for minicomputers and mainframe computers, but the budget voided all such considerations (American Farmland Trust, 1985). The system had to be microcomputer based. Yet, available software for developing a land information system was very scarce in 1984 (it is only slightly more available today). Thus, developing a workable system for a set of needs which were not fully understood at the time of the search and are constantly evolving is a very difficult task. Even the meaning of the terms of the field: Geographic Information Systems (GIS), Spatial Information Systems, and Geographic Data Bases, seems to be a matter of opinion on the part of those who develop them, thus talking to vendors is not very helpful. Peuquet (1977) provided a broad definition of a GIS in a paper eight year ago, yet some vendors seem
unaware of the effort. In this project, all agreed that a relational database was suitable to store attribute data for the land parcels. Yet, as the expense to support a GIS would be greater, the need was debated by a consideration of the needs of the PDC and its constituent governments.

Relational Databases as Land Information Systems

A relational database is one in which the data are arranged in a matrix. Each row is referred to as a record (or observation), a group of individual data items that are logically related together. Within each record are fields. A field contains one item concerning the record and is analogous to a column in the database (Ashton-Tate, 1984, p. 1-14). In an LIS application, the rows (records) of the matrix represent individual land units (of any desired description), and the columns (fields) represent various characteristics of those land units. For example, in a system based on land parcels, the records represent the different land parcels, and the fields represent various characteristics of those parcels such as ownership, zoning, municipality, planning area, assessed value, etc. (Figure 2).

<table>
<thead>
<tr>
<th>VARIABLES (fields - columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARCELS (records - rows)</td>
</tr>
<tr>
<td>PARCEL ID</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>N</td>
</tr>
</tbody>
</table>

Figure 2. A Relational Database Structure for LIS

This database structure makes certain operations very simple to accomplish. Searches for land parcels having particular characteristics are simple because the database maintains all information in consistent fields (columns). For example, all parcels over 10 acres in size that are zoned for agriculture may be identified by searching only two columns in the data file. For each parcel that meets the criteria, the parcel identifier can be printed giving a listing of all qualifying parcels. Because all the data values across a row refer to the same land unit, many fields can be combined in a single search. By their internal structures, relational databases can produce and process files of nearly unlimited numbers of records and more than 100 fields. Total file sizes may be
nearly as large as the mass storage devices used by the hardware system; they are not limited by the relatively smaller memories of microcomputers. The major advantage of a relational database is its simplicity. Adopted for land information, the system can provide some analysis capabilities, and can produce analyses very quickly. Due to its simple structure, it is easy to learn and use. It can be operated on very basic hardware requiring only a computer and a monitor (a printer is helpful but not required).

Geographic Information Systems as Land Information Systems

A Geographic Information System (GIS) is one in which the data to be analyzed are stored in a cartographic data structure (usually raster or vector). Maps each cover exactly the same ground area, and are referred to as "overlays", single-factor maps", "images", and "data elements," an element containing a map of one and only one particular thematic distribution (Berry, pp. 1-2). For example, if a GIS were to cover a county in the United States, one element might contain a map of political boundary information, another a map of the highway network, another of population density, and another of the land use for the county (Figure 3). Because the data are all stored as maps, two major functions of a GIS are the entry and display of maps. Other functions typically include varied techniques for comparing maps (elements) by overlay, and for analyzing individual maps. For example, an overlay of an element containing existing land use polygons upon an element containing proposed zoning polygons would yield a third element depicting the areas that would be in violation of the new ordinance (Figure 4). Analysis functions within single elements, either original elements or those created by the GIS during overlay, would minimally include the ability to measure areas. An area analysis function could be called to determine the number of acres that would be in violation of a new zoning ordinance by measuring the new element created during the overlay of existing land use and the proposed zoning ordinance. Additional analysis functions should include neighborhood searches in which distances are determined from specified map features. For example, one could determine the distance of each land parcel in a jurisdiction from the nearest shopping center. Some GIS packages contain far more analysis functions than others, thus care should be taken in deciding which functions are necessary for a proposed implementation.

A CARTOGRAPHIC VIEW OF THESE TWO LIS STRUCTURES

From the planner's standpoint, the system was to be used as an electronic file cabinet to support data entry, retrieval, and analysis. The problem with the file cabinet analogy is
that it is very limiting. Because data analysis is clearly a geographic process, a spatial-cartographic analysis of needs was also carried out during deliberations.

As originally envisioned by the committee, the system was strictly choroplethic. That is, the data were to be collected by previously defined units of land, the values to be collected for each parcel had to be considered homogeneous throughout the entire parcel, and the values had to be treated as discrete by parcel. For such a system, one might devise other data structures, but the microcomputer software marketplace offers many relational database structures. For most of the variables to be placed into the system, a relational database of choroplethic parcels is appropriate. Most of the variables are discrete and homogeneous within each parcel.

However, choroplethic structures have many weaknesses. First, choroplethic maps are the most generalized of all maps, often to the point of significant data loss. For example, a region of people having low income levels may be split among several enumeration units such that none of the units appear to have low income levels (Figure 5, after Sinton, 1977).
Data must be fit into land units that have been defined with no regard to any spatial components of the data. That is, the spatial structure of the enumeration procedure takes precedence over the spatial structure of the data themselves. Data values are discrete within each enumeration unit and as such values can change only at enumeration unit boundaries. Choroplethic structures are useful only for area data, thus the inclusion of data related to points or lines is not feasible in a single data file. Also, many analyses are very difficult to accomplish using strictly a relational approach because of a lack of spatial topology.

As an example of the ability of a choroplethic-relational structure to deal with a possible request, suppose that an industry becomes interested in locating in the Roanoke area. This industry is well aware of its needs for land, and presents the following criteria to the PDC for site evaluation:

1. The land must be at least five acres (2 hectares) in extent.
2. The land must be zoned for industry.
3. The land must be currently vacant or for sale.
4. The land must not be subject to flooding.
5. The land must not be over one mile from a heavy-duty highway allowing truck traffic.
6. The land must have no slope over ten percent.

How well could a relational database structure using parcel land units deal with this request? The question is one of simplicity (generalization) versus reality. It is possible to load any information one wishes into the columns of a relational structure, but one must be aware of the results of such decisions. A good understanding of the nature of the dimensions of spatial data (point, line, and area), and spatial data variation (discrete-continuous) is critical.

The first query is reasonably analyzed by a relational data structure. Available land is in parcel form, therefore areas for each parcel can be stored as items in the data base. Such a value is clearly discrete as it relates only to the parcel under consideration. The second query can also be answered well by a relational database providing that the zoning ordinance doesn't cross-cut the parcels such that there are several different zoning designations for various parcels. In the PDC database such was not the case, though some parcels were designated as mixed. The third query is also simple for the choroplethic-relational structure. A parcel is either vacant or not and is either for sale or not. This logical dichotomy is both homogeneous and discrete to each parcel. Query number four is not well handled by a relational structure. The concept of flooding is related to linear data (streams and rivers) not to area data, and the incidence of flooding is certainly not constrained by the locations of parcel boundaries. To generalize a flood plain to a land parcel system, one must determine whether a parcel is in a flood prone area or not. Clearly, for the sake of safety (and to avoid disastrous legal consequences) a parcel must be coded as in the flood plain if even a slight portion of it is subject to inundation. A relational search for parcels that are not in the floodplain may exclude some very good sites for this industry because of small portions of parcels that might flood. The generalization of these data is unrealistic and too crude to support a valid analysis.

The fifth query, the distance to the nearest major highway, is also inappropriate to be answered by a relational structure. As the structure is set up for the inclusion of information only on areas, the use of linear data is difficult. Certainly there is no way that the system could produce the distance estimate needed, as it does not contain
the spatial elements of topology or scale. Relying on manual measurements to create a field for the database is possible, but is incredibly inefficient. The PDC could hire staff to perform this task, yet their time would prove far more costly than the costs involved in designing a more flexible system. The final query involves additional spatial problems concerning both generalization and spatial variation, and is inappropriate for a relational database structure. Slope is based on elevation difference; elevation is a point phenomenon. The assignment of a single slope (or even several) to an entire parcel of land is a vast simplification of reality which may cause major inaccuracies in the database. Further, slope is a continuous as opposed to a discrete phenomenon, thus it cannot be described well by either a measure of extremes (as needed in this request) or of central tendency. Its entry into the system would have to rely on manual measurements which, again, would prove very costly to compile.

It is evident that a choroplethic-relational approach to this database is valid for some analyses, but very poor for others. For data that are truly discrete and homogeneous according to the parcel system, the relational database is not only appropriate but also optimal. It is a simple system with which to work, very quick to perform analyses, and is inexpensive to purchase. However, when the degree of generalization required to include certain variables becomes too great to be of use for many analyses, other structures such as a GIS should be considered to complement the relational structure.

To consider the GIS approach, let's look at the same example. First, the GIS would need to store up-to-date, accurate maps of five themes:

1. Land Parcel Boundaries
2. Zoning Boundaries
3. Streams
4. Major Highways
5. Elevation

All of the queries could be satisfied by the system. Area calculations can be accomplished by most GIS packages. Overlay of the zoning map on the parcel map could determine those parcels zoned for industry. The stream element could be combined with the elevation element to determine the flood-prone areas. These areas could then be overlaid on the parcel maps to look for areas in each potential industrial
site that are subject to flooding. The cartographic neighborhood search functions of the GIS could determine the distance from each major highway to each parcel, and the elevation element could be used to calculate slopes. The only difficulty might occur in tying the attribute values of vacancy/occupied and for sale/not for sale to the parcels, but this could be overcome by recoding the land parcel element in an appropriate manner.

Because GIS are generally far more flexible than realtional structures, the question in a GIS environment is not so often whether the system can satisfy a particular request, but whether the system should be used to answer particular queries. Three of the six queries in this example can be answered accurately and quickly by a relational database using far fewer computer resources than by a GIS. The other three are better answered using the GIS as it maintains a spatial topology, allows for multiple data dimensions (points, lines, and areas) to be included in an analysis, and has many different types of analyses available. It is clear that both structures could be used together more effectively than either one alone.

SOLUTIONS FOR THE FIFTH PLANNING DISTRICT COMMISSION

The PDC required that the system be available to answer a variety of queries. There was a strong sentiment that the system should not be as severely limited as the choroplethic-relational structure, thus both a relational database (dBASE III) and a GIS (AE-GIS) were purchased. It was felt that the GIS could be used to perform some analyses, and to provide data for the relational database. For example, though it is feasible with the GIS software, recalculation of the distance from each parcel to the nearest major highway every time a query requests the information is a waste of computer resources. As such values are appropriate for inclusion in the relational data base (i.e. values are discrete), the GIS can calculate the values once, and they can be stored in the relational database. The same procedure is useful for area measurements, and has been used for determination of dominant soil types by parcel.

Though the GIS and relational database packages do not communicate directly with each other, they are each able to produce data files on the system's hard disk which may be reformatted by in-house software routines for use by the other package. The Fifth PDC LIS integrates both a relational database and a GIS into a very workable product. The PDC is able to better serve its members by providing statistical analyses and geographic images in support of planning efforts. The interface between the relational
database and the GIS has helped to overcome many obstacles to the fulfillment of the initial goals. By using both a relational database and a GIS, the system has quick response to many sorts of requests for information through the relational database, yet it is not limited in its analysis capabilities to using only choropleth data.

REFERENCES


THE USE OF QUADTREES IN GEOGRAPHIC INFORMATION SYSTEMS
AND SPATIAL DATA HANDLING

David M. Mark
Department of Geography
University at Buffalo
Buffalo, New York 14260 U.S.A.

ABSTRACT

Quadtrees represent 2-dimensional (spatial) data in a way which takes advantage of spatial coherence in the phenomena being represented. A square enclosing a region of interest is divided into four quadrants, and these quadrants are recursively partitioned into subquadrants until all sub-quadrants are uniform with respect to the phenomenon. This paper explores the utility of quadtrees as a data structure for geographic information systems. Attention is focused on linear quadtrees, and in particular on 2-dimensional run-encoding, an effective structure for quadtrees.

INTRODUCTION

A Geographic Information System (GIS) is a computerized, spatially-referenced data base organized in such a way that spatial data input, analysis, and output may be accomplished. As is the case for any computer application, the issue of data structures is a critical one. Once a data structure for a GIS has been adopted, it is very difficult to change it; data structures also are a major factor in determining the efficiency with which queries can be answered within a GIS.

Recently, quadtrees have received considerable attention as a data structure for GIS applications (Mark and Lauzon, 1984; Peuquet, 1984; Samet et al., 1984; Lauzon et al., 1985). Quadtrees appear to have many advantages for handling coherent ('blocky') spatial data, but are inefficient for continuous surfaces such as topography. However, if quadtrees are to be used for natural resources GIS, it is essential to develop strategies for efficient integration of digital elevation model (DEM) data into the quadtree environment; such strategies will be discussed herein.

This paper presents a variety of issues related to a quadtree-based GIS, emphasizing: approaches for handling diverse spatial data types in a quadtree environment; strategies for covering very large areas; and applications of quadtrees and quadtree-related structures to problems in computational geometry, spatial search, and spatial modelling.
Quadtree Definitions and Basic Concepts
A quadtree is a spatial data structure based on a regular decomposition of an image into quadrants and subquadrants. This decomposition is conveniently represented by a rooted tree of out-degree 4 and depth n, where n is the resolution parameter. The level-0 nodes, or pixels, are conventionally taken as having a side length of one, and the full region has a side length of $2^n$. Each node of the tree corresponds to a sub-quadrant of the region. A polygon can be mapped onto the region by colouring the quadtree for the region (see Figure 1). By convention, a node is coloured BLACK if it lies wholly within the polygon, WHITE if wholly outside the polygon, and GRAY otherwise. Any sub-quadrant of level > 0 which contains areas of both colours is termed a GRAY node, and is divided into subquadrants.

![Quadtree Diagram](image)

Figure 1: A polygon (A) and its associated quadtree, represented as a tree (B).
Quadtree Data Structures

Early work on quadtrees represented relations among nodes by an explicit tree structure, with nodes linked to parents and children by pointers (see Samet, 1984, for a review). Another class of data structures, termed linear quadtrees (Gargantini, 1982; Mark and Lauzon, 1984; Samet et al., 1984; Mark and Abel, 1985; Samet and Tamminen, 1985) represents only leaf nodes, with those nodes identified by numeric keys. The form of these keys permits topologic and spatial relations to be determined from the key values through bit manipulations or modular arithmetic. The data structure is a list of the leaf nodes, in sequence by key.

Several space-efficient forms of linear quadtrees have been reported in the literature. Gargantini (1982) and Abel (1984) represent explicitly only the BLACK nodes of the tree, inferring WHITE nodes when required. Lauzon and Mark (Lauzon, 1983; Mark and Lauzon, 1984; Lauzon et al., 1985) have proposed a method termed 2-dimensional run-encoding (2DRE); in this approach, runs of nodes of the same colour which are consecutive in key sequence are represented by the last key within the run. For many algorithms of importance in geographic information systems applications (for example, overlay), 2DRE files can be used directly, without decoding. For other applications, the 2DRE file must be decoded into discrete quadtree leaves.

Mark and Lauzon (1985b) and Lauzon et al. (1985) have shown that 2DRE quadtrees usually require less space and fewer records than do other forms of linear quadtrees, which in turn are more space-efficient than pointer-based quadtrees. Samet (1984) correctly noted that space-efficiency is not always the most important measure of a data structure. However, overlay and certain other GIS algorithms can be solved in a time which is linear in the number of records stored; if a quadtree is represented by fewer records, it often will require less processing time.

ORDERING TWO-DIMENSIONAL SPACE

One of the most important issues in Geographic Information Systems (GIS) is this: geographic data are essentially 2-dimensional, whereas computer storage and processing are (thus far) essentially one-dimensional. No linear sequence can preserve all spatial properties of geographic data; however, the study of such 'orderings' of 2-dimensional space is an important yet neglected topic in GIS research.

An ordering of a set of N distinct spatial entities (polygons or points) may be defined as any one-to-one assignment of the integers 1 through N (or 0 through N-1) to the spatial entities. Thus, whereas the 2DRE data structure was originally developed as a compact representation of a
linear quadtree (Lauzon, 1983), 2DRE can also be viewed as an effective ordering of the pixels in a 2-D digital image. In fact, the overlay algorithm presented by Lauzon and Mark (Lauzon, 1983; Mark and Lauzon, 1984; Lauzon et al., 1985) will work for any ordering of an image, including raster (row-by-row), reversing raster (row-prime; Goodchild and Grandfield, 1983), and Hilbert-Peano (or pi; Goodchild and Grandfield, 1983) ordering, as well as the Morton (or N; White, 1983) ordering used in 2DRE.

One of the strengths of the 2DRE representation using the Morton sequence is its dual nature: it can be treated either as a 2-D ordering or as a compact linear quadtree. A disadvantage of the Morton sequence is that it does not traverse the image in a spatially-contiguous path. The Hilbert-Peano (pi) order may prove to be valuable for GIS applications, since, like row-prime, it always moves to a 4-neighbour, and yet, like Morton, pi-order visits all pixels in a quadtree subquadrant before leaving it. Goodchild and Grandfield (1983) found that images could be represented in fewer runs using pi-ordering than using Morton ordering. However, the recursion implicit in a quadtree is straightforward for Morton order and complicated for pi-order. The use of pi-order in GIS has yet to be explored.

DATA TYPES AND THEIR REPRESENTATIONS IN A QUADTREE-BASED GIS

It is proposed to recognize four fundamental classes of data, based mainly on dimensionality. For each quadtree area, several files may exist, each containing a different data type.

Type 0: Point Files
Point files would be held in Morton number. Point locations would be reported to the nearest pixel or grid cell and converted to within-patch pixel coordinates. The coordinates would then be combined to form the Morton number (key) of the point. Each record in the key-ordered file would contain the key of the point, its attributes and/or values, and perhaps its coordinates. Points would be held in key-order for ease of interfacing with other data types, and also because White (1983) and Abel and Smith (1984b) have shown that this is an effective data structure for solving closest point problems and other problems in computational geometry.

Type 1: Line Files
It is well-known that quadtrees do not handle line data well. However, a modified quadtree structure termed the nonminimal division quadtree has recently been proposed by Ayala et al. (1985); this may be the best method for
handling geographic line features such as power lines, roads, and streams. Note however that boundary lines would not be stored as line data; rather, the bounded regions are stored as coverage files (Type 2, below).

Type 2: Coverage Files
Coverage data, such as land use, soils, or rock types, will be stored as linear quadtree files. In most if not all cases, the 2-dimensional run-encoding method for file structuring is recommended for these space-filling coverages; the value of this structure already has been discussed above.

Type 3: Surface Files
Geographic information systems frequently involve natural resource data. In most such applications, elevation data (digital elevation models) represent an important component, both directly and in the form of derived measures such as slope. Thus, any data structure adopted for a resources GIS must be able to efficiently integrate DEM data with other geographic data.

Quadtree representations are not efficient if neighbouring cells seldom have identical values. This is, however, a characteristic of most DEMs, and of LANDSAT or other MSS satellite imagery. For such data, it is more efficient to store a value for every grid cell. For ease of interfacing with other data types, the proposed system would use the Morton number (key) of each pixel as a virtual address for referencing the elevation within a contiguous binary file (Lauzon et al., 1985; Cebrian et al., 1985). Cebrian et al. (1985) presented details of a strategy for integrating such data into a quadtree-based GIS, and for performing basic DEM analysis and display procedures.

Relating Type 2 (Thematic) to Type 3 (DEM) data
As noted above, the combination of Morton-order DEM files and 2DRE quadtree files provides the basis for efficient interfacing of the two types of data. Two different display procedures for illustrating associations between topography and other data have been devised and implemented (Cebrian et al., 1985). The first displays topographic properties (heights, slopes, slope aspects, etc.) within only those areas having some particular property or properties (such as a certain land cover or soil type) in one or more coverage files. The search area is defined by one or more 2DRE files. The second overlay procedure produces graphic output by overlaying thematic data on an image produced by analytical hill shading. In order to produce such a display, a special palette must be defined and loaded into the graphics device. This palette consists of several series of colours; each
series has a particular hue, but with different values (see Cebrian et al., 1985).

HANDLING VERY LARGE AREAS IN A QUADTREE-BASED GIS

Most real-world applications of GIS involve large areas. If a single coordinate system is used to cover such a very large area, either a large number of bits must be used for each coordinate, or precision must be lost. To avoid both of these undesirable effects, a GIS for a large area can partition the space into a set of mutually-exclusive and collectively-exhaustive patches (also known as frames). Such a system can be accommodated very easily within a quadtree-based GIS, since the same recursive system of quadrants can be used both within the patches and above them. Quadtrees for large areas do encounter problems when applied to sufficiently large portions of the Earth's surface, since squares cannot tessellate a sphere; thus, application of quadtrees to global or continental regions requires either the explicit use of map projections, or the modification of the quadtree concept to use quadrants which are not geometrically square.

Quadrangle-based Systems

One class of solutions to the problem involves partitioning the globe into areas which may be termed quadrangles, cells which are 'square' or 'rectangular' only in latitude-longitude terms. Such systems have a variety of analysis and display problems, since cells are not geometrically square, and furthermore change size and shape with latitude. However, the frequent use of latitude-longitude quadrangles for non-computerized mapping makes them attractive, since printed maps often are used as a primary data source for GIS.

A good example of a quadrangle-based GIS which is hierarchical in the quadtree sense (but only down to a certain scale) is the Canada Geographic Information System (Tomlinson et al., 1976). CGIS was the first full geographic information system. Indeed, although the quadtree idea is usually attributed to Klinger's 1971 paper, many quadtree-related concepts were included in CGIS in the mid-1960's (Morton, 1966; Tomlinson et al., 1976).

Systems Based on Map Projections

Another class of solutions maps the globe onto a plane or set of planes, using some map projection, and then defines a grid cell network in cartesian coordinates on the plane(s). Since no map projection can be both equal-area and conformal, square cells on the map would represent areas in the real world which vary in size, shape, or both. Mark and
Lauzon (1985a) proposed that a continental or world-wide scale GIS based on quadtree concepts should use the Universal Transverse Mercator (UTM) coordinate system. In such a system, three hierarchical levels would be used. The highest divides the world into UTM zones and subzones. Each UTM subzone is then divided into square patches, which are numbered according to the Morton. Finally, within each patch, a 256 by 256 array of cells is the basis for the quadtrees or other geographic data files.

For each patch, a variety of data sets may exist, using any or all of the four fundamental data types discussed above. The highest level of the GIS would consist of a data base management system (DBMS) which would contain a directory of patches, data types, and data sets actually available, with summary statistics relating to the file contents. In fact, the patches themselves can be treated as pixels, and summary statistics can be mapped at this highly generalized level.

The use of the UTM coordinate system is recommended for a quadtree-based GIS because: conversion of geographic (latitude-longitude) coordinates to UTM is well known and computer programs or formulae are readily available; UTM coordinates are in general use by the armed forces of the United States, Canada, the United Kingdom, and other countries; and the U.S. Geological Survey (USGS) distributes digital data either already in UTM coordinates or with coefficient for conversion to UTM contained in the file headers (see Elassal and Caruso, 1983; Allder and Elassal, 1984). For further details of the use of UTM coordinates in quadtree construction, see Mark and Lauzon (1985a).

QUADTREES, COMPUTATIONAL GEOMETRY, AND SPATIAL MODELLING

Computational geometry seeks efficient solutions to geometric problems. Many problems in spatial data handling stem from the fact that geographic space is 2-dimensional, whereas most computer processing is 1-dimensional (see discussion above). As an example, consider the problem of finding a point's nearest neighbour. To the human eye, candidates for the 'nearest neighbour' are obvious, and only a few measurements need be made to ascertain which point is indeed the closest. Limiting the amount of unnecessary searching is a central theme in computer handling of spatial data.

Quadtrees and related spatial data structures have considerable potential in this regard. Indeed, the region quadtree was first suggested (by Klinger, 1971) in the context of spatial search, and not as a data structure for images. Samet (1984, pp. 229-244) has provided a detailed review of methods for handling point data and rectangle data using quadtree-related methods. Abel and Smith (1983;
1984a) have discussed quadtree-based solutions to the rectangle retrieval and rectangle cover problems, which arise in applications as different as VLSI architecture and GIS. They also showed (Abel and Smith, 1984b) how linear quadtrees can be used for nearest neighbour calculations.

Quadtrees also hold particular promise for spatial modelling. For example, Mark (in prep.) shows how calculation of proximal (Thiessen) polygons can be simplified through the use of quadtrees. The basic algorithm is recursive: if all four corners of a square have a common closest point, then that square is part of the proximal polygon surrounding that point, and is part of the quadtree of the proximal polygon map. Otherwise, the current square is split into four sub-quadrants, and the algorithm tests each of these. The algorithm works because proximal polygons are convex. A recursive, quadtree-based approach may prove valuable in other spatial modelling situations.

SUMMARY

Quadtrees are very well-suited to many Geographic Information Systems (GIS) applications, chiefly because they represent 2-dimensional (spatial) data in a way which takes advantage of spatial coherence in the phenomenon being represented. This paper has emphasized the handling of diverse types of spatial data in a quadtree environment, strategies for covering very large areas, and the use of quadtrees and quadtree-related structures in computational geometry, spatial search, and spatial modelling.

ACKNOWLEDGEMENTS

This research has been supported by U. S. National Science Foundation Grant SES-8420789. Jean Paul Lauzon, Juan A. Cebrian, and James E. Mower have assisted with various aspects of the work presented in this paper.

REFERENCES


