A CONCEPTUAL FRAMEWORK AND COMPARISON OF SPATIAL DATA MODELS

DONNA J PEUQUET
University of California at Santa Barbara / California / United States

ABSTRACT This paper examines the major types of spatial data models currently known and places these models in a comprehensive framework. This framework is used to provide clarification of how varying data models, as well as their inherent advantages and disadvantages, are interrelated. It also provides an insight into how these conflicting demands may be balanced in a more systematic and predictable manner for practical applications, and reveals directions for needed future research.

1.0 INTRODUCTION

The rapidly expanding range of available spatial data in digital form, and the rapidly increasing need for their combined use, have revealed two very basic and severe problems associated with the application of automated spatial data handling technology: 1 a rigidity and narrowness in the range of applications and data types which can be accommodated, as well as, 2 unacceptable storage and speed efficiency for current and anticipated data volumes.

A general lack of versatility of spatial data processing systems exists, both for individual systems capabilities to accommodate a broader range of applications, as well as for the incorporation of differing types of spatial data from a variety of sources. The primary example of the need for very flexible spatial databases is the current attempts to incorporate LANDSAT and other remote sensed imagery and cartographic data within the same database. Spatial data have been accumulating at an increasingly rapid rate over the past two decades. This represents a very major investment and an extremely valuable resource which is in demand for a wide variety of research and decision making applications. Attempts to integrate these data into existing systems have, to date, proven extremely difficult, at best.

The problem of a lack of versatility and the difficulty of integration is compounded by the fact that current spatial databases are encountering severe problems with physical storage volumes and time needed for processing. The geographic database systems in existence, however, pale in comparison to the scope of the databases being actively planned by a number of federal agencies and private corporations. The U.S. Geological Survey is envisioning a cartographic database containing all information from 55,000 map sheets covering the entire United States. If these sheets were scanned once at, for example, 250 pixels per map inch (which is not high precision by cartographic standards), the total data would be approximately 1.5 × 10**15 pixels. Some common procedures on one of these digitized map sheets currently can take hours of computer time to execute. The USGS situation is, in turn, dwarfed by NASA's current plans for the development of a database incorporating all spacecraft data for the earth, as well as the other planets and bodies in our solar system.

These efficiency, versatility and integration problems are attributable in large part to the profound differences in the commonly used storage formats, and more basically, to a lack of fundamental knowledge concerning properties of spatial data and a lack of a unified body of knowledge on the design and evaluation of spatial data models.

This paper presents an overall taxonomy of digital data models for the storage and manipulation of geographic data and a review of selected data models within this taxonomic structure. This is intended to serve two purposes. The first is to provide a unified framework and some directions for continuing research in the area of spatial data handling techniques. The second is to help remedy the current state of confusion which seems to exist among practitioners as to the options and tradeoffs involved in this diverse subject.

This paper is organized in six sections. The first section provides a general introduction to the nature of current shortcomings of spatial data model technology in view of current and anticipated needs. This is followed by the presentation of a uniform theoretical framework, drawn primarily from the computer science literature. The third section reviews the various types of spatial data models as they are currently used in digital, geographic data storage and processing applications, with specific examples. The fourth section discusses recent developments in spatial data models. Here, changes in data model requirements are discussed within the context of recent research. The emphasis is placed on new approaches and on specific new models which hold promise but have not yet been used in any large-scale practical application. The fifth section briefly discusses the special problems involved with handling space-time data given the context of current theory and recent developments. The final section addresses future developments and their implications. Of necessity, this final section is broader in scope and deals with a number of developments which are affecting the demands on, and capabilities of, spatial databases in the future.

2.0 THEORETICAL FRAMEWORK

2.1 LEVELS OF DATA ABSTRACTION

A data model may be defined as a general description of specific sets of entities and the relationships between these sets of entities. An entity is a thing which exists and is distinguishable; i.e., we can tell one entity from another. Thus, a chair, a person and a lake are each an entity (Ullman, 1982). An entity set is a class of entities that possesses certain common characteristics. For example, lakes, mountains and desks are each entity sets. Relationships include such things as 'left of', 'less than' or 'parent of.' Both entities and relationships can have attributes, or properties. These associate a specific value from a domain of values for that attribute with each entity in an entity set. For example, a lake may have attributes of size, elevation and suspended particulates, among others.

A comparable definition of a data model was given by Codd (1981), who stated that a data model consists of three components; a collection of object types, a collection of operators and a collection of general integrity rules. As Date states, Codd was the first to formulate the concept of a data model in his original 1970 paper within the context of the relational database model (Codd, 1970). Date also asserts that: "The purpose of any data model, relational or otherwise, is of course to provide a formal means of representing information and a formal means of manipulating such a representation" (Date, 1983, pp. 182–183).

Since, as defined above, this is a human conceptualization and tends to be

DONNA J PEUQUET is an Associate Professor in the Department of Geography, University of California at Santa Barbara, California, USA. The research was sponsored by the National Aeronautics and Space Administration Information Sciences Branch, Grant No. NAC 5-369. MS *submitted August 1984*

CARTOGRAPHICA Vol 21 No 4 1984 pp 66–113

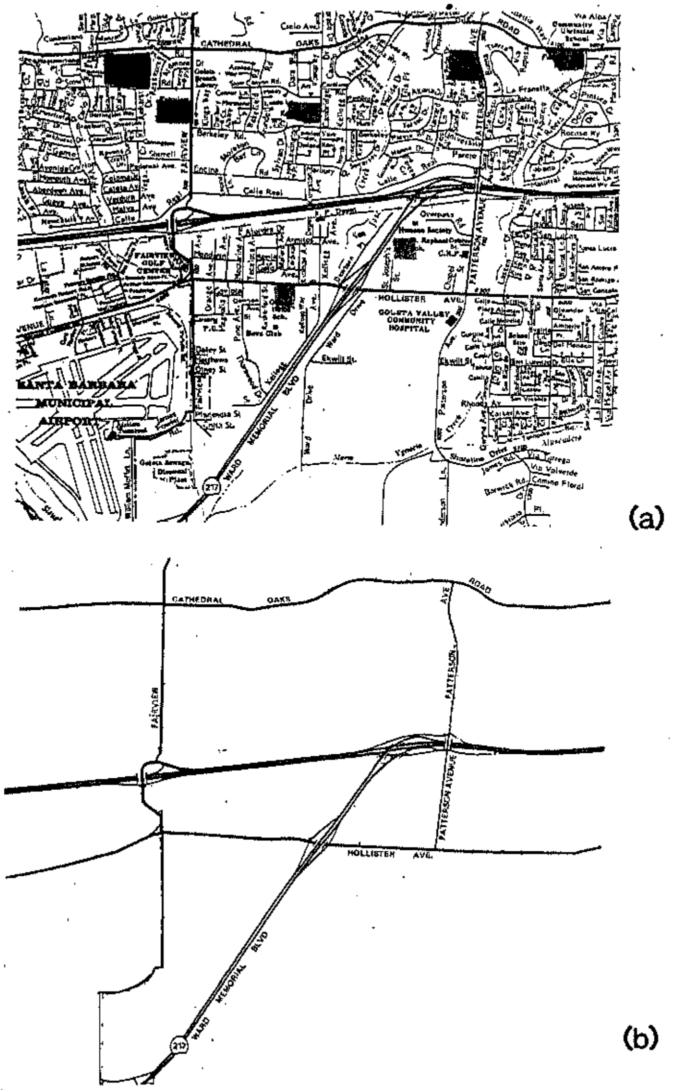


FIGURE 1. *The overall database model, as in* **a** *is likely to be confusing, overly complex for individual applications. Varying, simplified views of the data may be derived from the overall database model for specific applications, as in* **b**.

tailored to a given application, different users and different applications are likely to have different data models to represent the same phenomenon (Figure 1). As the word 'model' implies, the most basic characteristic of a data model is that it is an abstraction of reality. Each data model represents reality with a varying level of completeness.

Many data model designers realize that in order to determine how a collection of data is to be ultimately represented in digital form, the data need to be viewed at a number of levels. These levels progress from reality, through the abstract, user-oriented information structure, to the concrete, machine-oriented storage structure. There is, however, a lack of universal agreement as to how many levels of abstraction one should distinguish (Klinger, Fu & Kuni, 1977; Martin, 1975; Senko, 1973; Senko, et. al., 1976; Tompa, 1977; Nyerges, 1980). These differences can in large part be attributed to context. For the purposes of the present discussion, four levels will be utilized (Figure 2):

Reality – the phenomenon as it actually exists, including all aspects which may or may not be perceived by individuals;

Data Model – an abstraction of the real world which incorporates only those properties thought to be relevant to the application or applications at hand, usually a human conceptualization of reality;

Data Structure – a representation of the data model often expressed in terms of diagrams, lists and arrays designed to reflect the recording of the data in computer code;

File Structure – the representation of the data in storage hardware.

These last three views of data correspond to the major steps involved in database design and implementation. The overall process is one of progressively refining general statements into more specific statements. Within a level, the process of stepwise refinement would be used to provide a smooth transition from one level to the next.

The term 'data model' is used again here, but in the narrower context of a specific level of data abstraction. This is the result of a considerable amount of confusion which existed within the computer science, image processing, and geographic literature. The problem is a historical one. The term 'data structure' was commonly used as the generic term or used synonymously with 'data model'. However, with the development of systematic software design techniques and the easing of restrictions of the computing environment due to software and hardware technological advancements, the 'nuts and bolts' of language and hardware implementation is no longer a dominating force in database design. Thus, the term 'data model', in this context of levels of data abstraction, has evolved to connote a human conceptualization of reality, without consideration of hardware and other implementation conventions or restrictions.

A data structure is built upon the data model, and details the arrangement of the data elements. This can therefore be described as a structural model, with individual elements within each group organized into lists and arrays, and the relationships explicitly defined. This is equivalent to the mathematician's broad definition of a graph (Mark, 1979). Relationships between objects, or data elements, may be expressed explicitly or implicitly. Explicit relationships are written

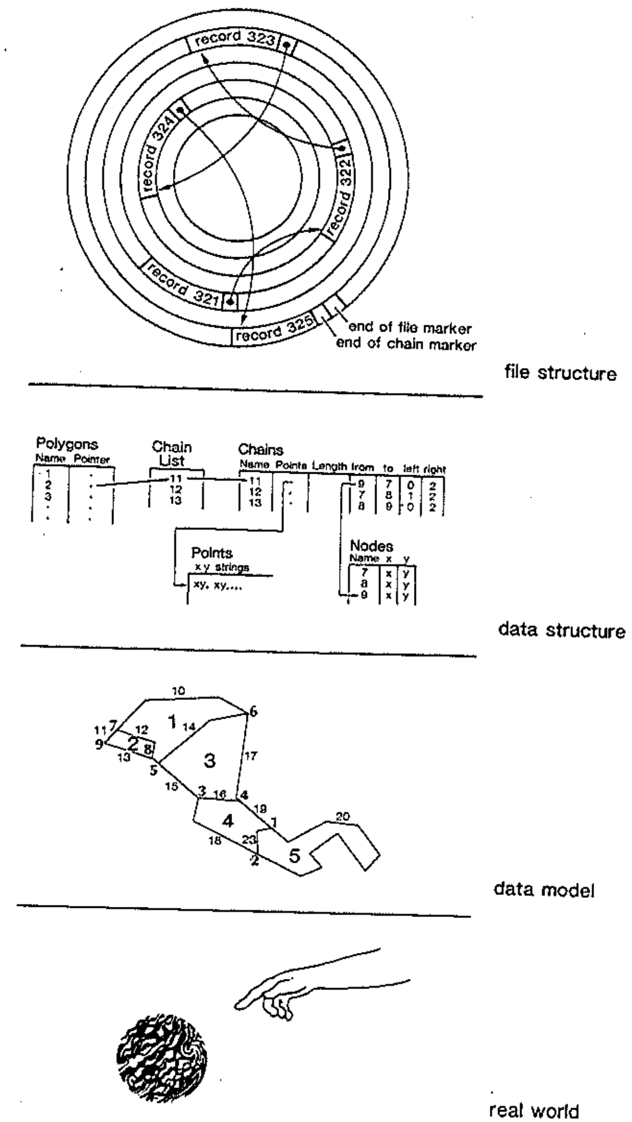


FIGURE 2. *Levels of data abstraction.*

into the data structure as data elements themselves. Implicit relationships can be indicated by the relative position of the individual data elements. Derivation of some implicit relationships may require computation through analysis of some or all of the data. An example would be nearest neighbor of a point among points distributed irregularly in space.

A file structure defines the physical implementation mechanism (i.e., the storage model). This is the translation of the data structure into a specific hardware/software environment.

2.2 GENERAL CONCEPTS

Since no model or abstraction of reality can represent all aspects of reality, it is impossible to design a general-purpose data model that is equally useful in all situations. This is particularly true when dealing with complex phenomena. For example, some spatial data models, when implemented in a digital environment, are good for plotting, but very inefficient for analytic purposes. Other data structures may be excellent for specific analytical processes, but may be extremely inefficient for producing graphics.

Varying approaches have been used in the design of spatial data models. To provide an example of the range of approaches which have been used, Bouillé's approach attempted to derive a data model which included all identifiable entities and their relationships into what he terms 'phenomenon-based design' in deriving the 'phenomenon structure' (Bouillé, 1978). Data models and subsequent data structures derived from such an approach, in attempting nearly complete representation of reality, tend to become like reality usually is – extremely complex. The result would most often be a level of complexity far beyond that which is useful or efficient in a computer context, and would contain many entities and relationships which are not essential to the application at hand.

Mark, on the other hand, adopts a philosophy that the data structure or data model design should be driven by its intended use and exclude any entities and relationships not relevant to that use (Mark, 1979). This results in a data model which tends to be a far from complete representation of reality, but instead contains only the essential elements necessary for a particular task. Such a minimalist approach, compared to the phenomenon-based design process of Bouillé, tends to produce models of minimum complexity.

These two views toward data model design represent two opposite extremes in the basic tradeoff involved in the data modelling process. The more perfectly a model represents reality, (i.e., the more completely all entities and possible relations are incorporated), the more robust and flexible that model will be in application. However, the more precisely the model fits a single application, excluding entities and relations not required to deal with that application, the more efficient it will tend to be in storage space commutation and ease of use.

The selection or design of a data model must, therefore, be based both on the nature of the phenomenon that the data represents and the specific manipulation processes which will be required to be performed on the data. This fact has been apparent to some degree to designers and builders of geographic data handling systems and geographic databases, however, the precise mechanisms of the tradeoffs involved between the various options available have never been discussed in depth.

The process of deriving an optimum balance between these two positions is best accomplished in practice by utilizing both of these approaches simultaneously in a 'both ends toward the middle' process. This is a process which has, unfortunately, not yet been formalized.

2.3 THE NATURE OF GEOGRAPHIC DATA

The term 'spatial' data applies to any data concerning phenomenon areally distributed in two-, three-, or N-dimensions. This includes such things as bubble chamber tracks in physics and engineering schematics. Geographic data, more specifically, are spatial data which normally refer to data pertaining to the earth. These may be two-dimensional, modelling the surface of the earth as a plane, or three-dimensional to describe subsurface or atmospheric phenomena. A fourth dimension could be added for time series data, as well. In the context of the present discussion, the term 'geographic' may also apply to data pertaining to other planets and objects in space.

There are several types of spatial data, and the differences between them become obvious when they are displayed in graphic form, as shown in Figure 3. The first is point data where each data element is associated with a single location in two- or three-dimensional space, such as the locations of cities of the United States. The second is line data. With this data type, the location is described by a string of spatial coordinates. These can represent either: **a** isolated lines where individual lines are not connected in any systematic manner, such as fault lines, **b** elements of tree structures, such as river systems, or **c** elements of network structures, as in the case of road systems.

The third type is polygon data, where the location of a data element is represented by a closed string of spatial coordinates. Polygon data are thus associated with areas over a defined space. These data can themselves be any one of three types: **a** isolated polygons, where the boundary of each polygon is not shared in any part by any other polygon, **b** adjacent polygons, where each polygon boundary segment is shared with at least one other polygon, and **c** nested polygons, where one or more polygons lie entirely within another polygon. An example of adjacent polygons is the state boundaries in a map of the United States. Contour lines on a topographic map are an example of nested polygon data.

A fourth category of data is some mixture of the above types. This might include different line structures mixed together, line structures mixed with a polygon structure or with discrete points. For example, in a map of the United States a state may be bounded by a river which is both a boundary between adjacent polygons as well as part of the tree structure of a river network. These four categories of spatial data are known as image or coordinate data (IGU, 1975; IGU, 1976). This means that these data portray the spatial locations and configurations of individual entities. A spatial data entity may be a point, line, polygon, or a combination of these. Each entity also has characteristics which

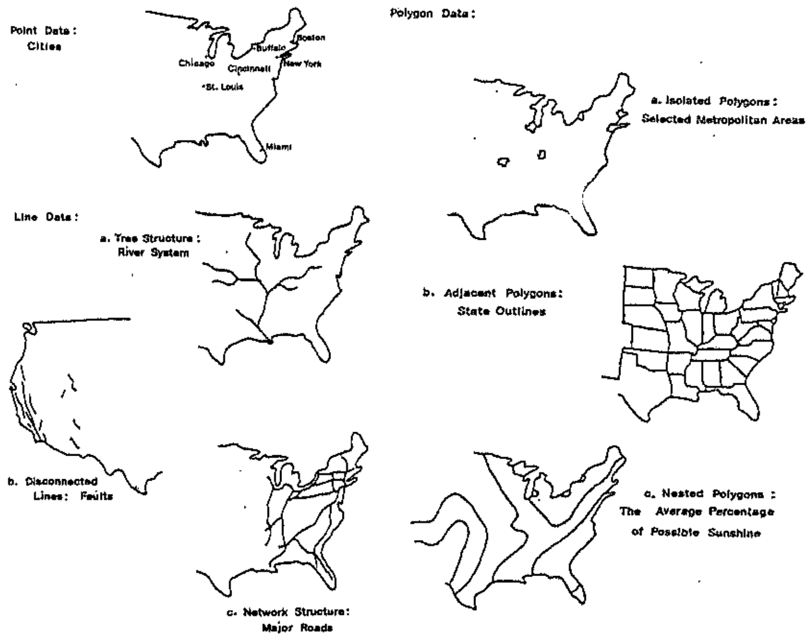


FIGURE 3. *Examples of spatial data types.*

describe it called attribute or descriptor data. For example, the latitude and longitude coordinates of the city of Santa Barbara are part of the image data set while its population would be part of the descriptor data set. Similarly, the coordinates which make up a spatial entity such as the outline of the State of California are image data, while statistics such as the total number of forested acres are descriptor data.

Spatial pheonomena, and spatial data models, have a number of characteristics which significantly differentiate them from one-dimensional or list-type models. First, spatial entities have individual, unique definitions which reflect the entities' location in space. For geographic data, these definitions are commonly very complex, given the tendency of natural phenomena to occur in irregular, complex patterns. Particularly for geographic data, these definitions are recorded in terms of a coordinate system. This coordinate system may be one of a number of types; latitude and longitude, UTM street address, etc. These coordinate systems may not necessarily have precise, mathematical transformations, such as street address to latitude and longitude.

The relationships between spatial entities are generally very numerous, and, in fact, given the nature of reality or our perceptions of it, and the limitations of the modeling process, it is normally impossible to store all of them. The definitions of these relationships, and the entities themselves in the case of geographic data, also tend to be inexact and context dependent. This is true of even very basic spatial relationships such as 'near' and 'far', or 'left' and 'right'.

The combination of these properties (multidimensionality, fuzzy entities and relationship definitions and complex spatial definitions) make the modeling of geographic data uniquely difficult. The models themselves tend to be complex and the resultant data files tend to be not very compact.

An additional problem arises in the transformation of a conceptual data model into data structure and file structure views for computer implementation. Graphic input devices, such as digitizers, transform area, line, and point structures into numeric, computer-readable form by recording spatial coordinates of map entities. There is a basic problem underlying this transformation. Spatial data are by definition two- or three-dimensional. How then can these data be represented in computer memory which is usually linear, or one-dimensional in nature, while preserving these implicit spatial interrelationships? If they are simply listed in a continuous linear stream, coordinates of the entities contain neither the topology inherent in line networks or adjacent polygons, nor spatial relationships, such as 'above' or 'left of'. These relationships are data in themselves and are often of primary importance, particularly to geographers, when examining spatial data (Dacey & Marble, 1965). The coordinates must therefore be structured so as to preserve these two- or three-dimensional relationships and yet be capable of being stored in linear or list fashion within the computer.

2.4 FORM VS. FUNCTION

The performance *vs.* representational fidelity tradeoff mentioned in section 2.2, impacts directly upon the storage, manipulative and retrieval characteristics of the data structure and physical file structure. It is necessary to examine these tradeoffs utilizing a specific set of usage-based criteria so that the overall quality or suitability of a specific data model can be evaluated within a particular context. The general criteria are:

1 completeness
2 robustness
3 versatility
4 efficiency
5 ease of generation.

Completeness may be thought of in terms of the proportion of all entities and relationships existing in reality which are represented in the model of a particular phenomenon. Robustness is the degree to which the data model can accommodate special circumstances or unusual instances, such as a polygon with a hole in it. Efficiency includes both compactness (storage efficiency) and speed of use (time efficiency). Ease of generation is the amount of effort needed to convert required data in some other form into the form required the data model.

In varying degree, each of these factors enters into consideration for any given application. The relative importance of each factor is a function of the particular type of data to be used and the overall operational requirements of the system. For example, if the database to be generated will be very large and must perform in an interactive context, compromises would likely be necessitated with the first three factors because overall efficiency and ease of generation would predominate.

It is possible to quantitatively measure the performance of several of these criteria, such as speed and space efficiency for a particular data model. It is not possible, however, to provide quantitative measures for the more abstract factors of data completeness, robustness or versatility. This, combined with the fact that we still have little knowledge of the performance characteristics of a wide range of spatial processing algorithms and how they interact with other algorithms and varying data models, indicates that the spatial data modeling process is much more an art than a science. Experience and intuition will remain primary factors in the interpretation of vague system requirements specifications and the construction of satisfactory data models, particularly for complete and integrated geographic information systems.

Additional comments on the process of balancing tradeoffs in spatial data model design will be made at the beginning of Section 4.

3.0 EXAMPLES OF TRADITIONAL GEOGRAPHIC DATA MODELS

3.1 BASIC TYPES

Geographic data have traditionally been presented for analysis by means of two-dimensional analog models known as maps (Board, 1967). The map has also provided a convenient method of spatial data storage for later visual retrieval and subsequent manual updating, measuring or other processing. In order to update a map or display results of any manual procedure performed on the data, a new map must be hand drawn or the old one modified by hand. This process is laborious and time-consuming, requiring both skill and precision on the part of the individual drafting the map.

Two other basic types of spatial data models have evolved for storing image data in digital form; vector and tessellation models (Figure 4). In the vector type of data model, the basic logical unit in a geographical context corresponds to a line on a map such as a contour line, river, street, area boundary or a segment of one of these. A series of x-y point locations along the line is recorded as the components of a single data record. Points can be represented in a vector data organization as lines of zero length (i.e., one x-y location). With the polygonal mesh type of organization, on the other hand, the basic logical unit is a single cell or unit of space in the mesh. These two types are thus logical duals of each other.

Common usage has usually considered the two basic spatial data model types to be raster, or grid, and vector. As this paper will show, however, the class of

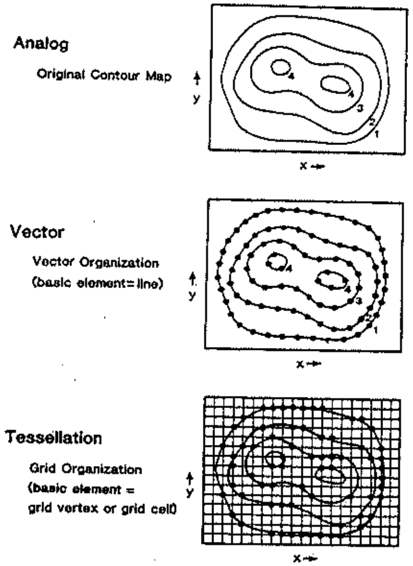


FIGURE 4. *Basic types of spatial data models.*

non-vector spatial data models encompasses much more than data models based on a rectangular or square mesh. This class includes any infinitely repeatable pattern of a regular polygon or polyhedron. The term used in geometry for this is a 'regular tessellation'. A tessellation in two dimensions is analogous to a mosaic, and in three dimensions to a honeycomb (Coxeter, 1973).

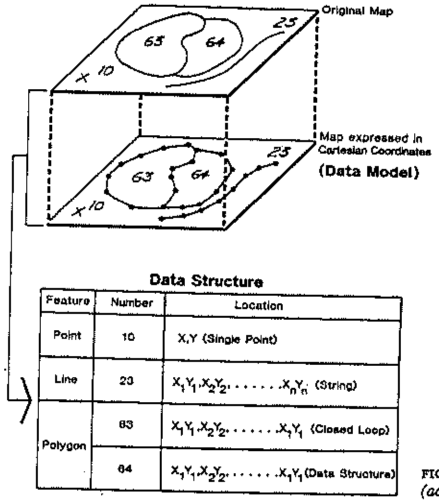
There also exists what can be viewed as a third type of spatial data model – the hybrid type. This class of data model is a recent development which possesses characteristics of both vector and tessellation data models.

Each of these approaches has also been used in fields other than geography to represent spatial data, such as scanner images in picture processing. The characteristics of each of these types of models and their tradeoffs for representing geographic phenomena should become clearer through the discussion of some specific examples of some 'classic' geographic data models.

3.2 VECTOR DATA MODELS

3.2.1 *Spaghetti Model*

The simplest vector data model for geographic data is a direct line-for-line translation of the paper map. As shown in Figure 5, each entity on the map becomes one logical record in the digital file, and is defined as strings of x-y coordinates. This structure is very simple and easy to understand since, in essence, the map remains the conceptual model and the x-y coordinate file is more precisely a data structure. The two-dimensional map model is translated into a list, or one-dimensional model. Although all entities are spatially defined, no spatial relationships are retained. Thus, a digital cartographic data file constructed in this manner is commonly referred to as a 'spaghetti file'; i.e., a collection of coordinate strings heaped together with no inherent structure. A polygon recorded in this manner is represented by a closed string of x-y coordinates which define its boundary. For adjacent polygon data, this results in recording the x-y coordinates of shared boundary segments twice – once for each polygon.



Data Structure

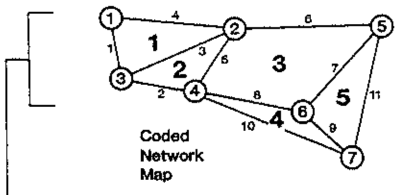
Feature	Number	Location
Point	10	X,Y (Single Point)
Line	23	$X_1Y_1,X_2Y_2,\ldots\ldots,X_nY_n$ (String)
Polygon	63	$X_1Y_1,X_2Y_2,\ldots\ldots,X_1Y_1$ (Closed Loop)
	64	$X_1Y_1,X_2Y_2,\ldots\ldots,X_1Y_1$ (Data Structure)

FIGURE 5. *The 'spaghetti' data model (adapted from Dangermond, 1982).*

The 'spaghetti' model is very inefficient for most types of spatial analyses, since any spatial relationships which are implicit in the original analog document must be derived through computation. Nevertheless, the lack of stored spatial relationships, which are extraneous to the plotting process, makes the spaghetti model efficient for reproducing the original graphic image. The spaghetti model is thus often used for applications that are limited to the simpler forms of computer-assisted cartographic production. Corrections and updates of the line data must rely on visual checks of graphic output.

3.2.2 *Topologic Model*

The most popular method of retaining spatial relationships among entities is to explicitly record adjacency information in what is known as a topologic data model. A simplified example of this is shown in Figure 6. Here, the basic logical entity is a straight line segment. A line segment begins or ends at the intersection with another line or at a bend in the line. Each individual line segment is recorded



Link #	Right Polygon	Left Polygon	Node 1	Node 2
1	1	0	3	1
2	2	0	4	3
3	2	1	3	2
4	1	0	1	2
5	3	2	4	2
6	3	0	2	5
7	5	3	5	6
8	4	3	6	4
9	5	4	7	6
10	4	0	7	4
11	0	5	5	7

Topologically Coded Network & Polygon File

Node #	X Coordinate	Y Coordinate
1	23	8
2	17	17
3	29	15
4	28	21
5	8	26
6	22	30
7	24	38

X,Y Coordinate Node File

FIGURE 6. *The Topological data model (from Dangermond, 1982).*

with the coordinates of its two endpoints. In addition, the identifier, or name of the polygons on either side of the line is recorded. In this way, the more elementary spatial relationships are explicitly retained and can be used for analysis. In addition, this topological information allows the spatial definitions of points, lines, and polygon-type entities to be stored in a non-redundant manner. This is particularly advantageous for adjacent polygons. As the example in Figure 6 shows, each line segment is recorded only once. The definitions and adjacency information for individual polygons are then defined by all individual line segments which comprise that polygon on the same side, either the right or the left.

3.2.3 GBF/DIME

The GBF/DIME (Geographic Base File/Dual Independent Map Encoding) model is the best known model built upon this topological concept. It was devised by the U.S. Census Bureau for digitally storing street maps to aid in the gathering and tabulation of Census data by providing geographically referenced address information in computerized form (U.S. Census, 1969). Developed as an improvement of the Address Coding Guides, the initial GBF/DIME files were created in the early 1970s.

In GBF/DIME file, each street, river, railroad line, municipal boundary, etc., is represented as a series of straight line segments. A straight line segment ends where two lines intersect or at the point a line changes direction. At these points and at line endpoints, nodes are identified (Figure 7).

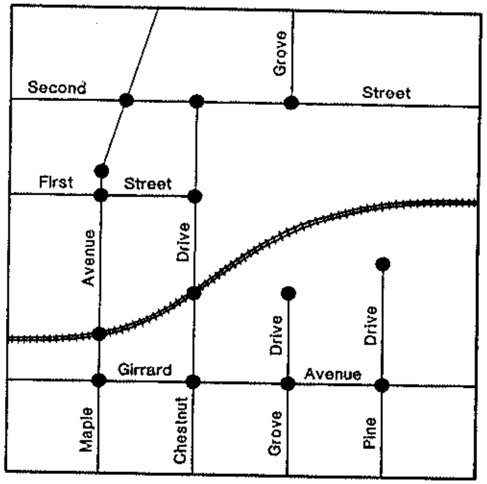
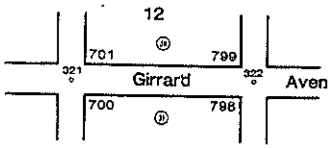


FIGURE 7. *Graphic elements of a DIME file.*



Street Name	Girrard
Street Type	Avenue
Left Addresses	701–799
Right Addresses	700–798
Left Block	38
Left Tract	12
Right Block	31
Right Tract	12
Low Node	321
X-Y Coordinate	155 000 – 232 000
High Node	322
X-Y Coordinate	156 000 – 234 000

FIGURE 8. *Contents of a sample DIME file record.*

As shown in Figure 8, each GBF/DIME line segment record contains Census tract and block identifiers for the polygons on each side. The DIME model offers a significant addition to the basic topological model in that it explicitly assigns a direction to each straight line segment by recording a From-node (i.e., low node) and a To-node (i.e., high node). The result is a directed graph which can be used to

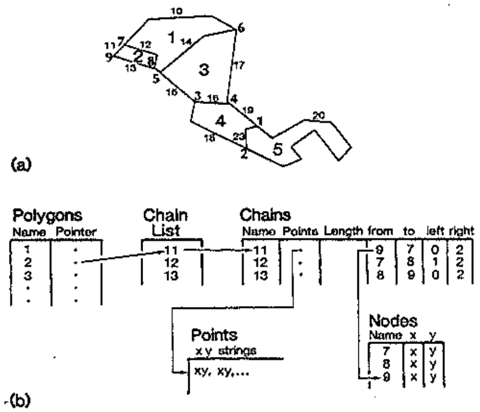


FIGURE 9. a *the* POLYVRT *data model;* b *the* POLYVRT *data structure.*

automatically check for missing segments and other errors in the file, by following the line segments which comprise the boundary of each census block (i.e., polygon) named in the file. This walk around each polygon is done by matching the To-node identifier of the current line segment with the From-node identifier of another line segment via a search of the file. If line segment records cannot be found to completely chain around a polygon in this manner, a line segment is missing or a node identifier is incorrect.

Another feature worth noting is that each line segment is spatially defined, according to the definition of the model, using both street addresses and UTM coordinates. This is in recognition of the fact that some locational systems (e.g., street addresses), which may be needed for some types of applications, cannot be directly derived from conventional cartesian or polar coordinate systems.

The main problem with the DIME model, like the previous two described, is that individual line segments do not occur in any particular sequence order. To retrieve any particular line segment a sequential, exhaustive search must be performed on the entire file. To retrieve all line segments which define the boundary of a polygon, an exhaustive search must be done as many times as there are line segments in the polygon boundary!

3.2.4 POLYVRT

POLYVRT (POLYgon conVERTer) was developed by Peucker and Chrisman (1975) and implemented at the Harvard Laboratory for Computer Graphics in the late 1970s. This model overcomes the very major retrieval inefficiencies seen in simpler topologic structures by explicitly and separately storing each type of data entity separately in a hierarchical data structure (Figure 9). To make the distinctions between types of entities both logically and topologically meaningful, a chain is denoted as the basic line entity. A chain is defined as a sequence of straight line segments which begins and ends at a node. A node is defined as the intersection point between two chains. The point coordinate information to define each chain is not stored as part of the chain record. Instead, a pointer to the beginning of this information within a separate Points file is recorded. Similarly, pointers are given within the Polygons file to the individual chains which comprise it. Note that the individual chain records contain the same explicit direction and topology information used within GBF/DIME; From- and To- nodes as well as the left and right adjacent polygons. If a chain defines an outer boundary of the entire area, such as for chain 13 in Figure 9, this outer area is denoted as polygon '0'.

This structure provides a number of advantages over GBF/DIME for retrieval and manipulation. First, the hierarchical structure allows selective retrieval of only specific classes of data at a time. A second advantage of the POLYVRT model is that queries concerning the adjacency of polygons need only deal with the polygon and chain portion of the data. Only the individual chains which bound the polygons of interest are retrieved. The actual coordinate definitions are not retrieved until explicitly needed for such operations as plotting or distance calculations.

The number of line or chain records in a POLYVRT data base depends only upon the number of polygons present in the data and not on the detail of their boundaries. In computer implementation, this physical separation allows a much greater efficiency in needed central memory space as well as speed for many operations. This gives POLYVRT a significant advantage for use with entities which have highly convoluted boundaries. However, this physical separation also causes the need for a link or pointer structure. These non-data elements add a significant amount of extra bulk to the model. The amount of overhead this generates usually cannot be tolerated for databases containing a large number of entities. The other major disadvantage is that incorrect pointers can be extremely difficult to detect or correct. The initial generation of this structure can also be cumbersome and time-consuming.

On the other hand, the POLYVRT approach has considerable versatility. Peucker and Chrisman represent a POLYVRT data model and its corresponding data structure which are tailored to represent a set of adjacent polygons. The model can also be augmented for the representation of more complex data. This does not violate the basic concept of the model to add another level to the hierarchy, such as an additional level of polygons. Using this modified POLYVRT to represent a map of the United States, for example, the higher polygons could be states and the lower polygons could represent counties.

Other changes to the basic POLYVRT structure open other possibilities. Various types of polygons in the same level of the hierarchy can be defined by a prefix added to the polygon identifier. Additional information concerning polygons, nodes and chains can be encoded into their respective identifiers in a similar manner.

Peucker and Chrisman discuss how the POLYVRT structure could be used for topographic data by encoding all ridges and channel lines as chains, and encoding the peaks, passes and pits as nodes. An auxiliary structure is used with this, representing a triangulated grid with the sample elevation points at the vertices of the triangle. This dual structure seems cumbersome but reveals still further

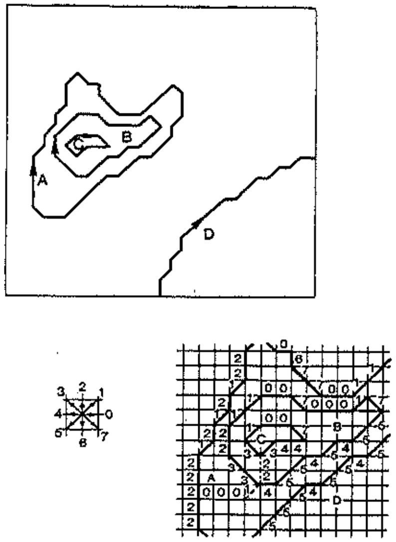


FIGURE 10. *Contour map and the resulting chain-coded lines (adapted from Freeman, 1974).*

possibilities for uses of a POLYVRT approach. It would be a great aid in overlaying two-dimensional with three-dimensional data if the two types were in the same or similar format.

3.2.5 Chaincodes

Chaincode approaches are actually a method of coordinate compaction rather than a data model. They are included in this discussion for two reasons. First, this methodology provides significant enhancements in compaction and analytic capabilities and therefore has been frequently integrated into spatial data models, including some which will be discussed below. Second, chaincodes have had a major impact on spatial data models and spatial data processing to such an extent that they are commonly viewed as a data model in its own right.

The classical chaincoding approach is known as Freeman-Hoffman chaincodes (Freeman, 1974). This consists of assigning a unique directional code between 0–7 for each of eight unit-length vectors as shown in Figure 10. The eight directions include the cardinal compass directions, plus the diagonals. Using this scheme to encode line data upon a grid of given unit resolution results in a very compact digital representation. As also seen in this example, x-y coordinate information need only be recorded for the beginning of each line. Direction is inherent in this scheme, providing an additional compaction advantage for portraying directed data, such as stream or road networks.

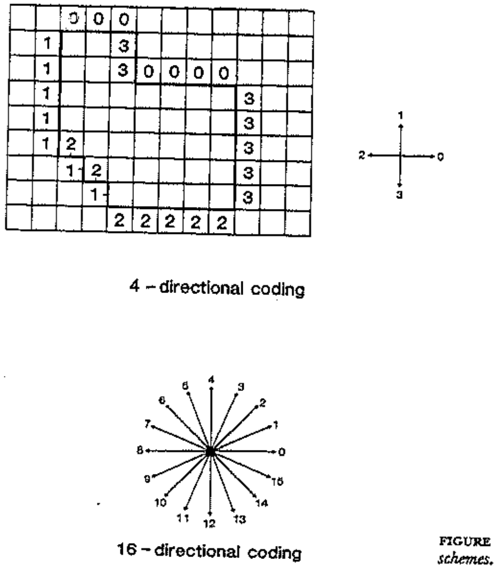


FIGURE 11. *4- and 6-directional chain-coding schemes.*

Through the use of special code sequences, special topological situations such as line intersections can be noted. One of the special coding sequences is also used for providing a mechanism for run-length encoding. This eliminates the need for repeated direction codes for long, straight lines. The flag used to signal that one of these special codes follows is '04'. This directional chaincode sequence would mean that the line retraces itself, a meaningless sequence in most cases. It thus can be used as a convention[illegible] the reader is referred to Freeman for his complete listing [illegible] course, be augmented or changed to suit a particular application.

There have been several variations of this coding scheme derived. The first, also described by Freeman (1979) is to utilize a 4, 16 or 32 vector notation on the same square lattice. The 4-directional encoding scheme allows representation of each code with 2 instead of 3 bits, and is sufficient in cases where the data tend to consist of long lines which are orthogonal to one another such as in some engineering applications. Sixteen or 32-direction coding allows for more accurate encoding of arbitrary-shaped curves. This smooths out the staircasing effect introduced by the directional approximations necessary when fewer directions are used for encoding (Figure 11). Similarly, there is a direct relationship between the number of directional vectors and the unit vector length for any given desired encoding accuracy for arbitrarily-shaped lines. In terms of compaction, this obviously

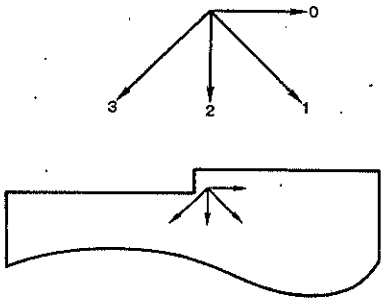


FIGURE 12. *The Raster Chaincoding scheme (adapted from Cederberg, 1979).*

presents a tradeoff between the number of direction-vector codes required to represent a given line and the number of bits required to represent each code.

The second well-known variation on the Freeman-Hoffman chaincoding scheme is Raster Chaincodes, or RC codes, introduced by Cederberg (Cederberg, 1979). This scheme uses only half of the standard 8 direction vectors as shown in Figure 12. This was designed to process scan-line-formatted data in raster order (each scan line in sequence, top to bottom and left to right) to produce chaincoded vector-formatted data. Since processing in this order never encounters 'backwards' vectors relative to the processing direction, only half of the eight standard direction codes are needed. This restricted directionality does, however, have the effect of segmenting the directional continuity of arbitrary shaped lines. If directional continuity of vector data is needed, the conversion of raster chaincodes to Freeman-Hoffman chaincodes is a straightforward process of 'flipping' or reversing the directionality of selected vector segments. For closed polygons, the selection of vector segments to be reversed is based on the Jacobsen Plumbline algorithm. This conversion process was described in detail by Chakravarty (1981).

A third variation on the chaincoding concept is its use on a hexagonal rather than a square lattice (Scholten and Wilson, 1983).

The primary disadvantage of chaincodes is that no spatial relationships are retained. It is, in fact, a compact spaghetti-format notation. Another disadvantage is that coordinate transformations, particularly rotation, are more difficult with chaincoded data.

As previously mentioned, the primary advantage of the chaincoding approach is its compactness. Chaincoding schemes are frequently incorporated into other schemes for the purpose of combining the compaction advantage of chaincodes with the advantages of another data model. The use of incremental directional codes instead of cartesian coordinates results in better performance characteristics than the simple spaghetti data model. The standard method of operation for vector plotters is to draw via sequences of short line segments utilizing (usually) 8 possible direction vectors. Vector plotter hardware, thus seems to be tailormade for chaincoded data. Graphic output on these devices requires no coordinate translation, making the process very efficient.

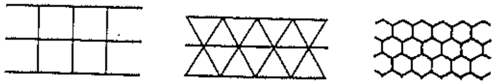


FIGURE 13. *The three regular tessellations.*

The use of unit vector direction codes is also advantageous for a number of measurement and analytical procedures, such as distance calculations and shape analyses. Algorithms for many of these procedures for chaincoded data were developed and documented by Freeman (1974; 1979).

3.3 TESSELLATION MODELS

As stated in the beginning of this section, tessellation, or polygonal mesh models, represent the logical dual of the vector approach. Individual entities become the basic data units for which spatial information is explicitly recorded in vector models. With tessellation models, on the other hand, the basic data unit is a unit of space for which entity information is explicitly recorded.

3.3.1 *Grid and other regular tessellations*

All three possible types of regular tessellations have been used as the basis of spatial data models. Each has differing functional characteristics which are based on the differing geometries of the elemental polygon (Ahuja, 1983). These three are square, triangular and hexagonal meshes (Figure 13).

Of these, the regular square mesh has historically been the most widely used primarily for two very practical reasons: 1 it is compatible with the array data structure built into the FORTRAN programming language, and 2 it is compatible with a number of different types of hardware devices used for spatial data capture and output. Fortunately, a number of higher-level computing languages are currently available which provide a great deal of flexibility in representing data through both additional intrinsic structures and user-defined structures. The ability to easily mix languages within the same program has also facilitated the programming task in general.

In the earliest days of computer cartography, the only graphic output device commonly available was the line printer (Tobler, 1959). Each character position on the line of print was viewed as a cell in a rectangular grid. Later devices for graphic input and output, particularly those designed for high speed, high volume operation, process data in rectangular mesh form. These include raster scanners, also known as mass digitizing devices, and color refresh CRTS. Remote sensing devices, such as the LANDSAT MSS capture data in gridded form as well (Peuquet and Boyle, 1984).

The tremendous data volumes being accumulated through the use of these grid-oriented, data input devices is in itself generating significant inertia toward using data in that form, rather than converting it to vector form.

The primary advantage of the regular hexagonal mesh is that all neighboring cells of a given cell are equidistant from that cell's centerpoint. Radial symmetry makes this model advantageous for radial search and retrieval functions. This is

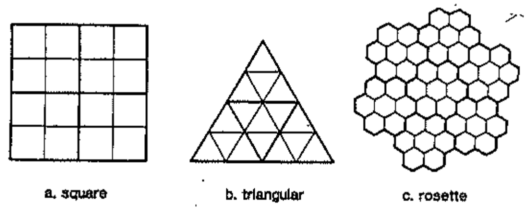


FIGURE 14. *The three regular tessellations in recursively subdivided form.*

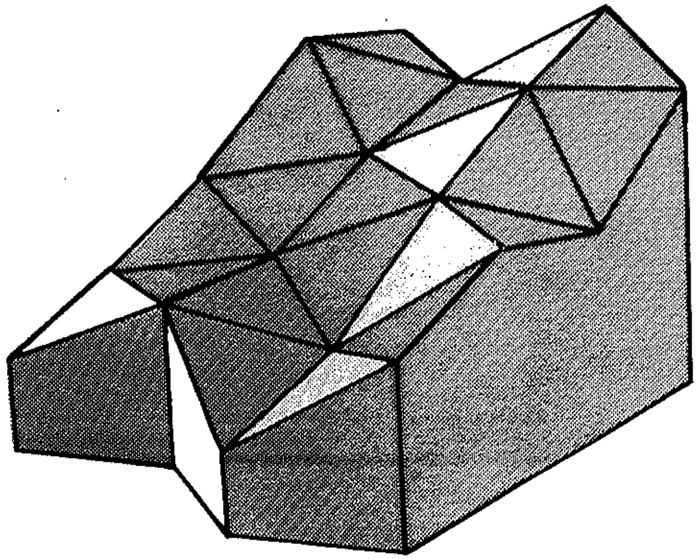


FIGURE 15. *A regular triangulated network representing surface data (adapted from Bengtsson and Nordbeck, 1964).*

unlike the square mesh where diagonal neighbors are not the same distance away as neighbors in the four cardinal directions from a central point.

A characteristic unique to all triangular tessellations, regular or irregular, is that the triangles do not all have the same orientation. This makes many procedures involving single-cell comparison operations which are simple to perform on the other two tessellations, much more complex. Nevertheless, this same characteristic gives triangular tessellations a unique advantage in representing terrain and other types of surface data. This is done by assigning a z-value to each vertex point in the regular triangular mesh (Figure 15). The triangular faces themselves can represent the same data via the assignment of slope and direction values.

Regular triangular meshes, however, are rarely used for representation of this type of data. Irregular triangular meshes are used instead, although Bengtsson and Nordbeck have shown that the interpolation of isarithms or contours is much easier and more consistent given a regular mesh (Bengtsson and Nordbeck, 1964). Perhaps a contributing factor in the almost total lack of use of the regular triangular mesh for surface data is simply that such data are normally not captured in a regular spatial sampling pattern. An irregular triangular mesh has a number of other advantages which will be discussed later in this paper.

In terms of processing efficiency on general procedures to compute spatial properties such as area and centroid calculations, or to perform spatial manipulations such as overlay and windowing, the algorithms initially devised for operation on square grids can easily be modified to work in the case of a triangular or hexagonal mesh. These, in fact, have the same order of computational complexity (Ahuja, 1983).

3.3.2 Nested tessellation models

Regular square and triangular meshes, as described above, can each be subdivided into smaller cells of the same shape, as shown in Figure 14. The critical difference between square, triangular and hexagonal tessellations on the plane is that only the square grid can be recursively subdivided with the areas of both the same shape and orientation. Triangles can be subdivided into other triangles, but the orientation problem remains. Hexagons cannot be subdivided into other hexagons, although the basic shape is approximated. These hexagonal 'rosettes' have ragged edges (Figure 13). Ahuja describes these geometrical differences in detail (Ahuja, 1983).

There are several very important advantages of a regular, recursive tessellation of the plane as a spatial data model. As a result, this particular type of data model is currently receiving a great deal of attention within the Computer Science community for a growing range of spatial data applications (Samet, 1984). The most studied and utilized of these models is the quadtree, based on the recursive decomposition of a grid (Figure 16).

The advantages of a quadtree model for geographical phenomena in addition to the advantages of a basic standard model include:

1 recursive subdivision of space in this manner functionally results in a regular, balanced tree structure of degree 4. This is a hierarchical, or tree, data model where each node has 4 sons. Tree storage and search techniques is one of the more thoroughly researched and better understood topics in computer science. Techniques are well documented for implementation of trees as a file structure, including compaction techniques and efficient addressing schemes.

2 In cartographic terms, this is a variable scale scheme based on powers of 2 and is compatible with conventional cartesian coordinate systems. This means that scale changes between these built-in scales merely require retrieving stored data at a lower or higher level in the tree. Stored data at multiple scales also can be used to get around problems of automated map generalization. The obvious cost of these features, however, is increased storage volume.

3 The recursive subdivision facilitates physically distributed storage, and greatly

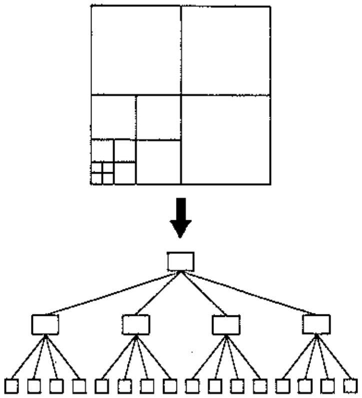


FIGURE 16. *The quadtree data model.*

facilitates browsing operations. Windowing, if designed to coincide with areas represented by quadtree cells, is also very efficient. These are features which are very advantageous for handling a very large database.

Advantages 1 and 3 also hold for the other two types of tessellations, taking into consideration that a recursive hexagonal tessellation has a branching factor of 7 instead of 4. Although all recursive tessellations can be viewed as having the variable scale property, the triangular and hexagonal versions do not have direct compatibility with cartesian coordinate systems.

The following is a brief discussion of the major types of quadtrees. A comprehensive discussion of quadtrees and all of its variant forms, as well as an extensive bibliography, has been given by Samet (1984).

Besides the general data model described above, the term quadtree has also acquired a generic meaning, to signify the entire class of hierarchical data structures which are based on the principle of recursive decomposition, many of which were developed in parallel. The 'true', or region quadtree was first described within the context of a spatial data model by Klinger (Klinger, 1971; Klinger and Dyer, 1976), who used the term Q-tree. Hunter was the first to use the term quadtree in this context (Hunter, 1978). Finkel and Bentley used a similar partition of space into rectangular quadrants (Finkel and Bentley, 1974). This model divides space based on the location of ordered points, rather than regular spatial decomposition (Figure 17). Although this was also originally termed a quadtree, it has become known as a point quadtree in order to avoid confusion. It is an adaptation of the binary search tree for two-dimensions (Knuth, 1975).

A data model related to the quadtree is the pyramid, which was developed within the field of image understanding (Tanimoto and Pavlidis, 1975). A pyramid is an exponentially tapering stack of discrete arrays, each one 1/4 the size of the previous without the explicit interlevel links of a tree structure. Because the pyramid does not have a strictly recursive structure, scales based on other than powers of two can be defined.

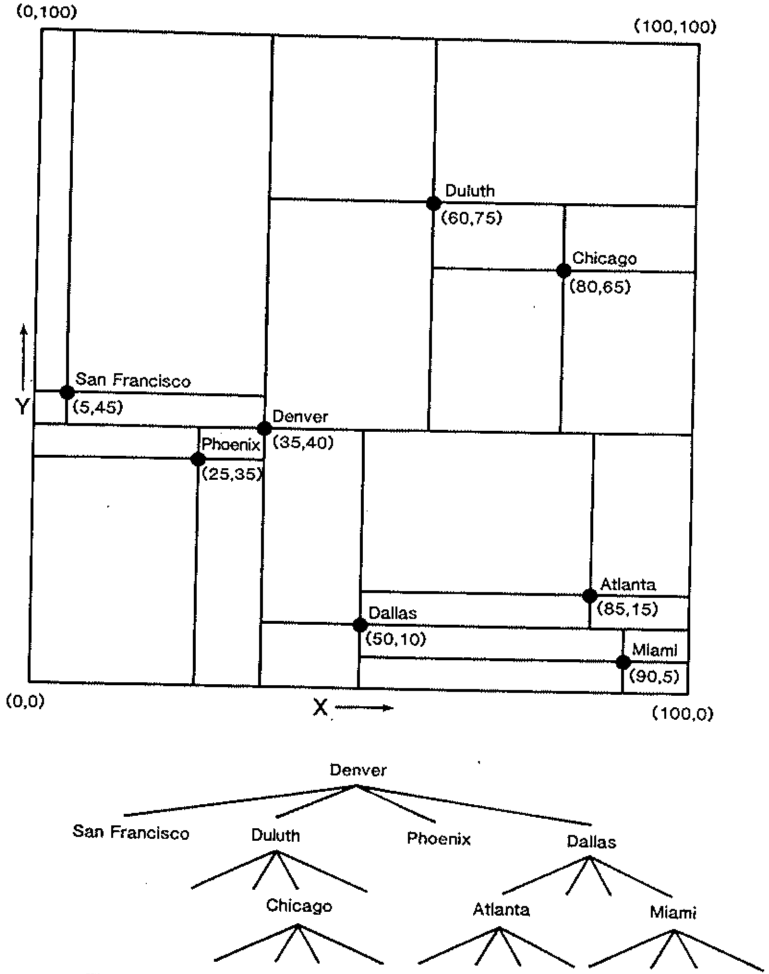


FIGURE 17. *The point quadtree data model (from Samet, 1984).*

3.3.2.1 Area quadtrees. The quadtree concept and all derivative algorithms may be extended into multiple dimensions (Reddy and Rubin, 1978; Jackins and Tanimoto, 1980; Jackins and Tanimoto, 1983). The oct-tree (branching factor = 8) or three-dimensional quadtree is probably the best known of these. Individual quadtrees representing different classes of data can also be spatially registered to form

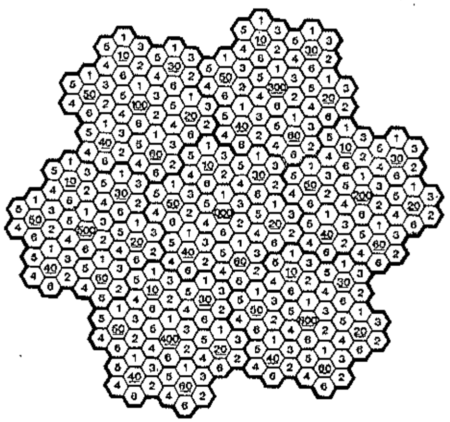


FIGURE 18. *A nested hexagonal tessellation with a hierarchical, base 7 indexing scheme.*

multiple layers, as can be done in a gridded database. This is known as a 'forest' of quadtrees.

The recursive decomposition based on the hexagonal tessellation, or septrees (branching factor = 7), retains the deficiency that a hexagon cannot be subdivided into smaller hexagons. This means that the smallest hexagonal resolution unit in a given implementation must be predetermined. Conversely, higher-level resolution units formed by an aggregation of hexagons can only approximate a hexagon (Figure 14). Algorithms for septrees have been developed by Gibson and Lucas (1982). This work has capitalized on the radial symmetry of hexagonal tessellation by basing these procedures on a base 7 addressing scheme, which they named Generalized Balanced Ternary, or GBT (Figure 18). Vectors, distance measurements and several other procedures can be performed directly on the GBF addresses without conversion.

Recursive decomposition based on the triangular tessellation is the other alternative. This model is called a triangular quadtree since each triangle is subdivided into four smaller triangles, yielding a tree with a branching factor of four. Again, this model retains all of the inherent advantages and disadvantages of the regular triangular tessellation with the added advantages associated with a hierarchical structure. Although a direct addressing scheme analogous to those for square and hexagonal tessellations is easily derived, such a scheme would not have any advantage in addition to allowing direct retrieval of individual stored data elements.

Generally, most developmental work on quadtree-type data models and associated algorithms has been based on classical tree storage and traversal techniques which are based on pointers. The alternative of using direct addressing techniques has been explored by a number of researchers in addition to Lucas (Abel and Smith, 1983; Gargantini, 1982). To distinguish these from the pointer-based approach, they have been termed linear quadtrees. This term is derived from the fact that by utilizing direct addressing structures, the data can be physically organized in linear fashion; i.e., as a list.

3.3.2.2 Point quadtrees. As stated above, point quadtrees base the subdivision on the location of ordered data points rather than regular spatial decomposition. A point quadtree takes one data point as the root and divides the area into quadrants based on this point (Figure 17). This is done recursively for each ordered data point, resulting in a tree of degree 4. Since the arrangement of data points in the tree is determined by relative location among the points, yielding a regular data decomposition rather than a regular areal decomposition, they are useful in applications which involve search and nearest neighbor operations.

One disadvantage to point quadtrees is that the shape of the tree is highly dependent on the order in which the points are added. Additions and deletions are therefore impossible except at the leaves of the tree.

A problem with multiple dimensions with any type of quadtree structure is that the branching factor becomes very large (i.e., 2k for k dimensions), which in turn would require much storage space. The k-d tree of Bentley is an improvement on the point quadtree by avoiding a large branching factor (Bentley, 1975). The k-d tree divides the area into two parts instead of four at each point, yielding a tree of degree 2 (Figure 19). The direction of this division is rotated among the coordinates for successive levels of the tree. Thus in the two-dimensional case, the data space could be divided in the x direction at even levels and the y direction at odd levels.

3.3.3 Irregular tessellations

There are a number of cases in which an irregular tessellation holds some advantages. The four most commonly used types for geographical data applications are square, triangular and variable (i.e., Thiessen) polygon meshes. The basic advantage of an irregular mesh is that the need for redundant data is eliminated and the structure of the mesh itself can be tailored to the areal distribution of the data. This scheme is a variable resolution model in the sense that the size and density of the elemental polygons vary over space.

An irregular mesh can be adjusted to reflect the density of data occurrences within each area of space. Thus, each cell can be defined as containing the same number of occurrences. The result is that cells become larger where data are sparse, and small where data are dense.

The fact that the size, shape, and orientation of the cells is a reflection of the size, shape, and orientation of the data elements themselves is also very useful for visual inspection of various types of analyses.

Perhaps the irregular tessellation most frequently used as a spatial data model is the triangulated irregular network (TIN) (Figure 20a). TINs are a standard method of representing terrain data for landform analysis, hill shading and hydrological applications. There are three primary reasons for this. First, it avoids the 'saddle point problem' which sometimes arises when drawing isopleths based on a square grid (Mark, 1975). Second, it facilitates the calculation of slope and

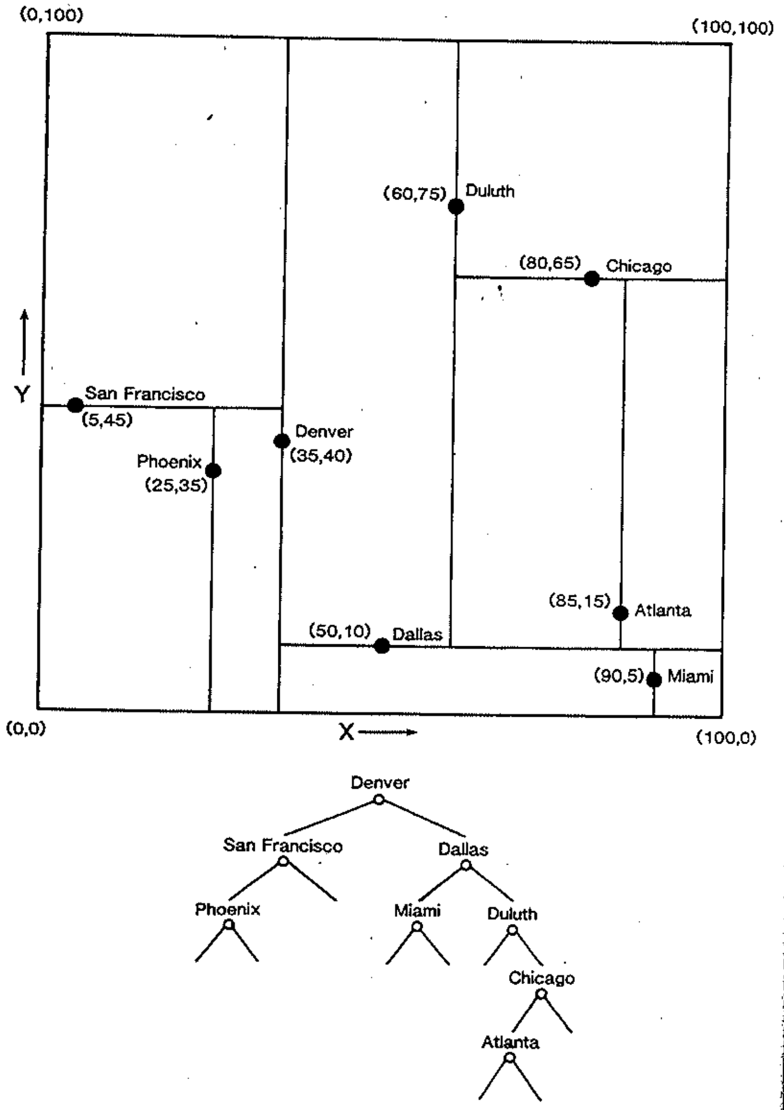


FIGURE 19. *The K-d tree data model (from Samet, 1984).*

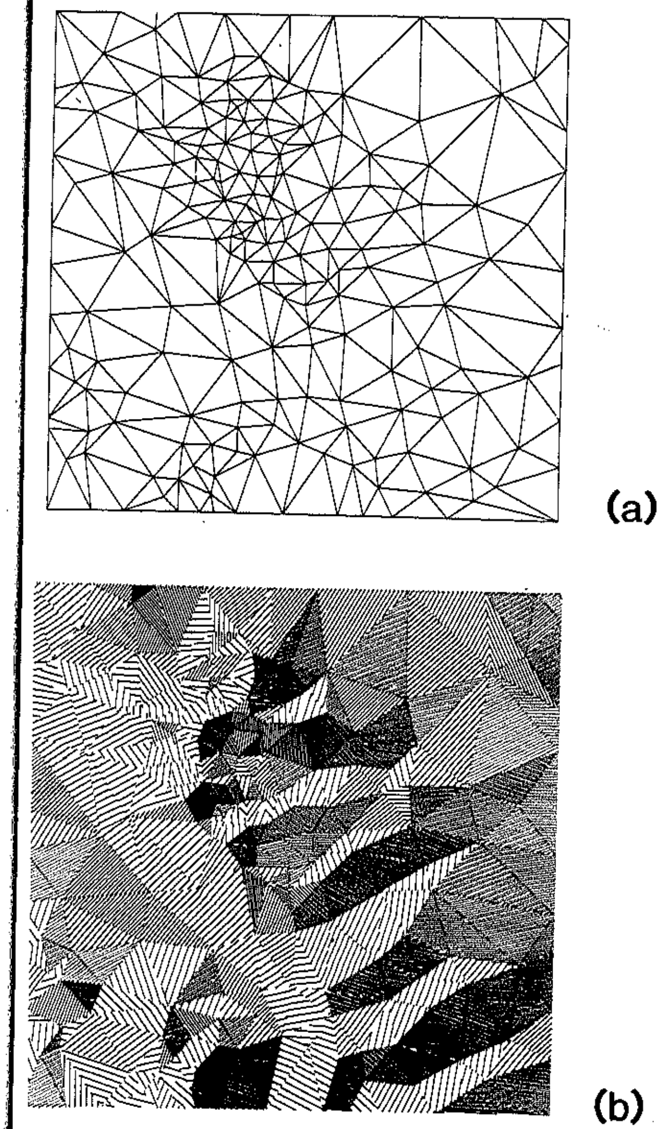


FIGURE 20. *A Triangulated irregular network* (TIN).

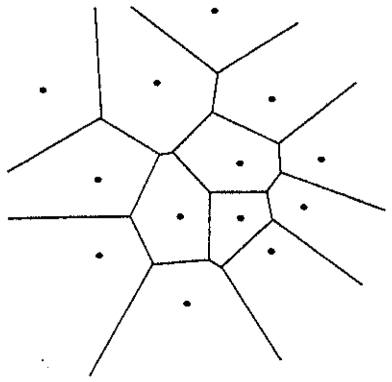


FIGURE 21. *An example of Thiessen polygons.*

other terrain-specific parameters. Third, the data are normally recorded at points distributed irregularly in space.

A major problem associated with irregular triangulated networks is that there are many possible different triangulations which can be generated from the same point set. There are thus also many different triangulation algorithms. Any triangulation algorithm will also require significantly more time than subdivision of a regularly spaced point set.

Thiessen polygons, also called Voronoi diagrams or dirichlet tessellations, are the logical dual of the irregular triangulated mesh. Thiessen polygons are constructed by bisecting the side of each triangle at a 90° angle, the result, as shown in Figure 21, is an irregular polygonal mesh where the polygons are convex and have a variable number of sides.

Rhynsburger has described the following alternate logical derivation (Rhynsburger, 1973). Given a finite number of distinct points that are at least three in number and distributed in some manner on a bounded plane, each point begins to propagate a circle at a constant rate. This growth continues until the boundary of a circle encounters another circle or the boundary of the plane. The analytical derivation of Thiessen polygons has been studied by a number of people (Rhynsburger, 1973; Kopec, 1963; Shamos, 1977).

Thiessen polygons are useful for efficient calculation in a range of adjacency, proximity and reachability analyses. These include closest point problems, smallest enclosing circle (Shamos, 1977), the 'post office' problem (Knuth, 1973) and others.

The first documented practical application of Thiessen polygons was in the determination of recipitation averages over drainage basins by Thiessen (1911), for whom Thiessen polygons were later named (Rhynsburger, 1973).

Two extensions of the basic concept have also been developed. The first of these is to assign a positive weight to each of the points which represents the point's power to influence its surrounding area, to produce a weighted Voronoi diagram. This was described by Boots (1979) and has particular advantages for marketing and facility location siting problems. Drysdale and Lee (1978) have also generalized the Voronoi diagram to handle disjoint line segments, intersecting line segments, circles, polygons and other geometric figures.

Although it is seen that various irregular polygonal tessellations are each uniquely suited to a particular type of data and set of analytical procedures they are very ill-suited for most other spatial manipulation and analytical tasks. For example, overlaying two irregular meshes is extremely difficult, at best. Generating irregular tessellations is also a complex and time-consuming task. These two factors make irregular tesselations unsuitable as database data models except in a few specialized applications.

3.3.4 Scan-line models

The parallel scan-line model, or raster, is a special case of the square mesh. The critical difference with the parallel scan-line model is that the cells are organized into single, contiguous rows across the data surface, usually in the x direction, but do not necessarily have coherence in the other direction. This is often the result of some form of compaction, such as raster run-length encoding. This is a format commonly used by 'mass digitizing' devices, such as the Scitex drum scanner.

Although this model is more compact than the square grid, it has many limitations for processing. Algorithms which are linear or parallel in nature (i.e., input to a process to be performed on individual cells does not include results of the same process for neighboring cells) can be performed on data in scan-line form with no extra computational burden in contrast to gridded data. This is because null cells (i.e., cells containing no data) must also be processed in the uncompacted, gridded form. Many procedures used in image processing fall into this category. Other processes which do depend upon neighborhood effects, require that scan-line data be converted into grid form.

3.3.5 Peano scans

A family of curves which generate a track through space in such a way that n-dimensional space is transformed into a line and vice versa was discovered in 1890 by the mathematician, Giuseppe Peano (Peano, 1973). These curves, also known as space-filling curves, preserve some of the spatial associativity of the scanned dataspace on the single dimension formed by the scan. Figure 22a shows an example of a simple two-dimensional Peano curve. With this particular version, all changes of direction are right angles. Figure 22b shows a similar Peano scan in three-dimensions.

Peano scans possess several properties which can be useful in some spatial data handling applications. These were summarized by Stevens, Lehar and Preston (1983):

1 the unbroken curve passes once through every locational element in the dataspace.
2 points close to each other in the curve are close to each other in space, and vice versa.
3 the curve acts as a transform to and from itself and n-dimensional space.

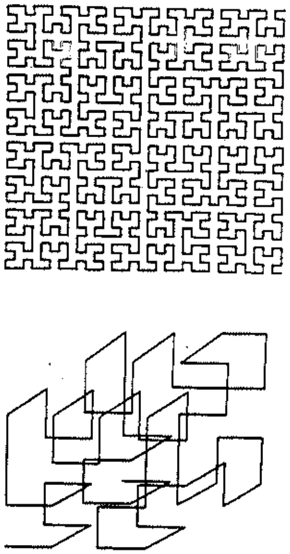
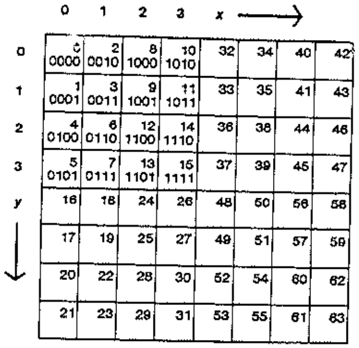


FIGURE 22. *A right-angle Peano curve in two and three dimensions (from Stevens, Zehar and Preston, 1983).*

The first known practical application of Peano curves as a digital geographic data model was as the areal indexing scheme within the Canada Geographic Information System (CGIS) (Tomlinson, 1973). This database divided gridded areal data into 'unit frames'. The frame size was determined for convenience of retrieval and processing. Each unit frame in the system was assigned a unique number starting at the origin of their coordinate system. From that point, frames are sequenced so that they fan out from the origin as frame number increases. This arrangement is shown in Figure 23 for the first 64 unit frames. This numbering scheme, named the Morton matrix after its designer (Morton, 1966), is in reality the trace of an Z-shaped (or N-shaped) Peano scan (Figure 24). This spatial indexing scheme utilizes the property that areas close together on the earth will likely have a minimum separation in a sequential digital file. This has the effect of reducing search time, especially for small areas. An additional benefit of this addressing structure is that Morton matrix addresses can be directly computed by interleaving the binary representation of the geographic x and y coordinates (Figure 25). This addressing scheme was examined more fully by White (1982) and Torf (1981). This work was built upon the recursive properties of Peano curves and the direct correspondence between the Z-shaped Peano curve and quadtree structures. It should also be noted that the quadtree addressing scheme used by Abel and Smith (1983) and Mark and Lauzon (1983) has one important difference from the Morton matrix scheme. The Morton matrix scheme is level-specific. In other words, the quadtree level cannot be determined from the address code itself. This level must be explicitly indicated by an additional code. This is not the case with the scheme used by Abel and Smith, in which the digits of the addresses correspond to the levels of the quadtree model (Figure 26).



	0	1	2	3	x →			
0	0 0000	2 0010	8 1000	10 1010	32	34	40	42
1	1 0001	3 0011	9 1001	11 1011	33	35	41	43
2	4 0100	6 0110	12 1100	14 1110	36	38	44	46
3	5 0101	7 0111	13 1101	15 1111	37	39	45	47
y	16	18	24	26	48	50	56	58
↓	17	19	25	27	49	51	57	59
	20	22	28	30	52	54	60	62
	21	23	29	31	53	55	61	63

FIGURE 23. *The morton matrix indexing scheme.*

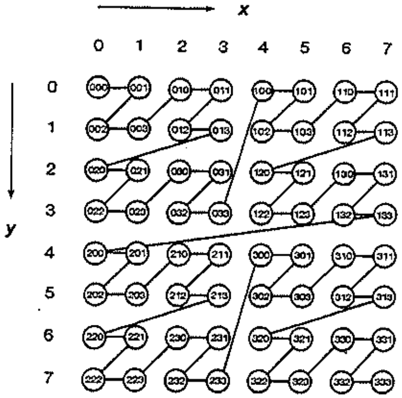


FIGURE 24. *The relationship between the morton matrix indexing scheme and the z-shaped Peano curve.*

The properties of Peano curves have also been shown to have significant utility for image processing applications (Stevens, 1983). These applications include data compression in the spatial and spectral domain, histogram equalization, adaptive thresholding and multispectral image display. The reason for this is that, as stated above, these and many other techniques for manipulating and analyzing imagery data are sequential and single channel operations, i.e., they are linear in nature. Peano curves can thus be used to collapse these multidimensional data into a single dimension. The property of preserving some of the spatial relationships in the one-dimensional Peano scan data allows improved interpretation, and thus improved results, from these procedures.

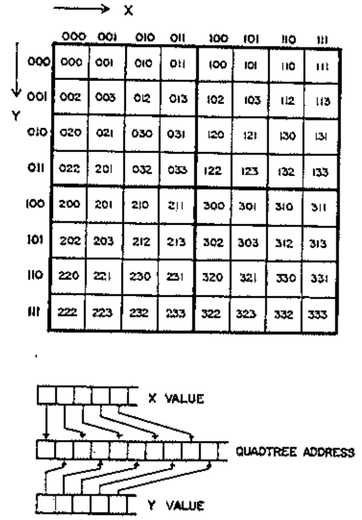


FIGURE 25. *Bitwise interlaced indexing scheme.*

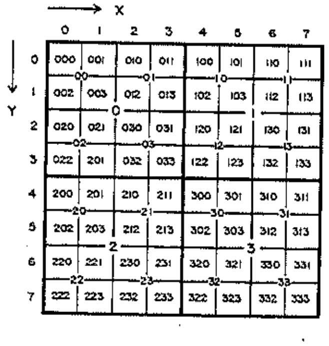


FIGURE 26. *Hierarchical, base 5 quadtree indexing scheme.*

3.4 RELATIVE MERITS

In summary, each of the three basic types of data models (paper map, vector, tessellation) have advantages and disadvantages which are inherent in the model itself. Individual models, such as the ones discussed above, can overcome these only to a limited degree, and always only by some sort of tradeoff. Vector data models are direct digital translations of the lines on a paper map. This means that the algorithms also tend to be direct translations of traditional manual methods.

The repertoire of vector-mode algorithms is thus both well-developed and familiar. The primary drawback of vector-type data models is that spatial relationships must be either explicitly recorded or computed. Since there is an infinite number of potential spatial relationships, this means that the essential relationships needed for a particular application or range of applications must be anticipated.

Conversely, spatial interrelationships are 'built-in' for regular, tessellation-type data models. Grid and raster data models are also compatible with modern high-speed graphic input and output devices. The primary drawback is that they tend to be not very compact. Regular tessellations tend to force the storage of redundant data values. Redundant data values can be avoided by the use of a wide variety of compaction techniques. Another drawback is that the algorithm repertoire is less fully developed. It is assumed that this latter drawback will diminish or disappear as the current increase in the use of raster and other tessellation-type models continues (Peuquet, 1979).

From a modeling perspective, vector and tessellation data models are logical duals of each other. The basic logical component of a vector model is a spatial entity, which may be identifiable on the ground or created with the context of a particular application. These may thus include lakes, rivers, roads, and entities such as 'the 20-foot contour level'. The spatial organization of these objects is explicitly stored as attributes of these objects. Conversely, the basic logical component of a tessellation model is a location in space. The existence of a given object at that location is explicitly stored as a locational attribute.

From this perspective, one can clearly see that neither type of data model is intrinsically a better representation of space. The representational and algorithmic advantages of each are data and application dependent, even though both theoretically have the capability to accommodate any type of data or procedure.

4.0 RECENT DEVELOPMENTS IN SPATIAL DATA MODELS

4.1 THE PROBLEMS OF VERY LARGE DATABASES AND DATA INTERCHANGE

As stated in the beginning of this paper, current and anticipated spatial data volumes have generated a two-faceted problem:

1 existing data structures are too inefficient and inflexible to meet current requirements; and

2 format conversions between different data structures to satisfy the current range of required applications produces significant processing overhead.

The rate of increase in data volumes and demands for fast performance has meant that storage and speed advances in computing hardware technology can no longer be relied upon to provide a cost-effective solution. Even if this brute-force approach were economically feasible, the amount of inefficiency often present with 'traditional' data structures represents unneccessary overhead which may, to a large degree, be alleviated by further developing our knowledge of spatial data models. This would at least result in reduced overall costs for spatial data handling on a practical level, and a cleaner solution from a theoretical standpoint.

As the size of any database becomes very large, several important problems arise which must be dealt with:

1 efficiency
2 heterogeneity
3 accuracy, and
4 security.

Dealing with these problems in order to maintain a functional database is critical, since a large database always represents a large investment of time and resources, and often is an integral part of the day-to-day operation of the owner organization.

Inefficiencies, even major and obvious ones, can often be tolerated if the database is small or infrequently used. In these cases, it is often more cost-effective to absorb the extra time and computing costs than to bear the expense of careful initial construction or retroactive fixing of a system. For any type of large database, however, overall space and time efficiency becomes a critical factor. Even in a governmental context where internal and external use of the database often is not expected to be self-supporting, inefficiencies in a large and frequently used system tend to multiply into a major drain of resources.

The problem in obtaining efficiency is that, as stated in the beginning of this paper, the current state-of-the-art does not allow optimally efficient spatial databases to be built in a predictable manner. Very little is known about the performance characteristics of many individual spatial data models and algorithms. Even less is known about how to combine groups of algorithms and data models in a complex system for optimal performance.

The problem of data heterogeneity becomes a frequent and major consideration in dealing with large databases and is the cause of most data interchange problems. The usual need in the earth sciences is to combine different types of data, from various organizational sources, captured through varying equipment and techniques, for varying purposes and to varying quality standards and resolutions.

Combining different types of data with different spatial resolutions can be achieved through the use of some of the new spatial data modeling approaches such as quadtrees. Dealing with variability with the other factors can only be done via coordination and standardization among the various data capture organizations and user groups. The overall accuracy and error characteristics of a database resulting from the combination of a number of data layers of differing error characteristics overlaid on top of each other is little understood, and most often impossible to calculate. What is understood, however, is that the overall error rate is multiplicative rather than additive. This problem could be significantly eased through improved documentation and coordination.

4.2 HYBRID VECTOR / TESSELLATION MODELS

One approach to the storage and processing tradeoffs between tessellation and vector data structures is to store the spatial data in (usually) raster or grid form, perhaps with only minor modification from its raw scanner raster output form. The data are then converted to vector format when advantageous for performing a given analytic or manipulative process. Frequently the result is then converted back again for graphic output. This conversion approach is the most commonly used because it is conceptually so straightforward. What is soon discovered, however, is that these data structure conversions can quickly become a bottleneck within a system as the volume of data and frequency of use increases (Peuquet, 1981a; Peuquet, 1981b). Tesselation-to-vector conversion requires some type of intricate line-following procedure, because cartographic lines are characteristically both convoluted and topologically complex. These conversion procedures represent significant system overhead which must be avoided, or at least minimized.

Another approach is to develop new tessellation or specifically raster-oriented algorithms for processes which currently have only vector-oriented solutions. Theoretically, this could eventually eliminate the need for vector data structures and result in the development of exclusively tessellation-oriented geographic information systems. This would be particularly desirable in applications where raster-formatted areal data, such as LANDSAT imagery, is used in conjunction with map line data.

A wide variety of efficient raster-oriented data analysis and manipulation procedures have already been developed, primarily within the field of image processing (Peuquet, 1979). However, some spatial analytical processes seem to be intrinsically sequential or vector-oriented and cannot be restated in a parallel manner (Lee, 1980). These include some commonly performed analytical procedures, such as network shortest-path and optimal-routing problems. Other processes may prove to be so much more efficient when performed in vector mode that the additional time overhead for data structure conversion would be canceled out. For these large-volume raster-based systems, the high overhead of 'forced' vectorization of data could thus be greatly reduced but not totally eliminated in a raster-oriented geographic information system if any intrinsically vector-oriented procedures are required by the user.

A possible solution to this dilemma is the development and use of hybrid types of spatial data models for geographic data which incorporate characteristics of both structures. The Vaster data model outlined below is the first such hybrid model and was developed by Peuquet (1981). This was developed as a raster-vector hybrid because of the prevalence of this particular data model dilemma, although it could be easily modified for hexagonal or triangular tessellations.

4.2.1 *The Vaster data model*

The basic logical unit of the raster-type model is the scan line, whereas a map line segment is the basic logical unit of the vector-type model. A map line segment is customarily defined to be a single uninterrupted map line. A line interruption is defined as the occurrence of the end of a map line, or the intersection with another line or the map boundary.

In contrast, the basic logical unit of the Vaster structure is the swath. Each swath spans a constant, known range over y and would correspond to a group of

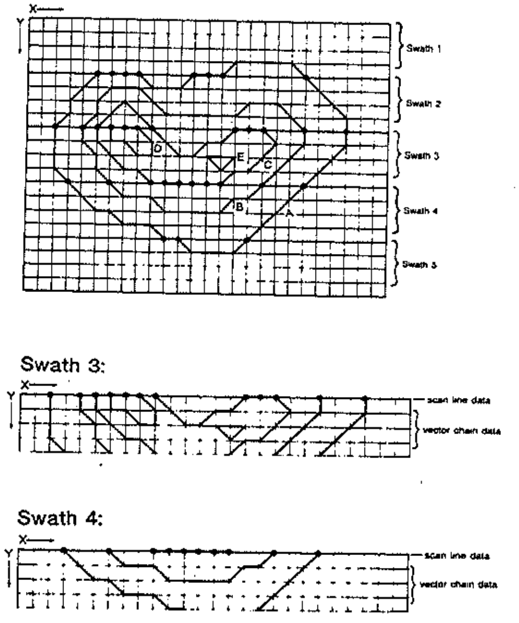
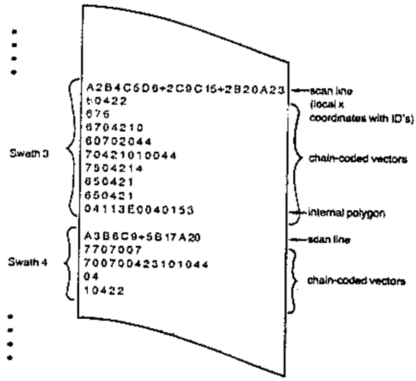


FIGURE 27. a *Vaster Organization (logical record-swath)*; b *Individual swaths.*

contiguous scan lines if the data were organized in raster format. Each swath contains a raster component and a vector component, as depicted in Figures 27a and 27b. Both components are recorded at the same grid resolution. The leading edge of each swath (minimum y value) is recorded in raster format as a single scan line and functions as the index record for the swath, containing an identifier and x-coordinate for each map-line intercept. The encoding used is similar to the structure developed by Merrill (1974). The raster encoding scheme used in the Vaster structure, however, contains all map line intersections within the same record. This is done to allow for efficient linkages between the raster and vector portions of the swath and to allow types of line structures other than nested polygons. The data contained in the remainder of the swath are recorded in vector format. All vectors contained in each swath are arranged in scan line intercept sequence in order of ascending x; polygons internal to a swath are listed separately. Each line intercept noted in the index-record functions as the end-point of each vector line segment within the swath, in sequence. Note that there is no scan line record at the end of the swath, since the next scan line record is functionally the leading edge of the next swath. In digital form, the raster record in each swath contains an ordered sequence of line identifier – x coordinate pairs.



```
A2B4C6D6+2C9C15+2B20A23
60422
676
6704210
60702044
70421010044
7504214
650421
650421
04113E0040153
A3B6C9+5B17A20
7707007
70070042310104 4
04
10422
```

FIGURE 28. *Vaster-formatted data in digital form.*

Each map line in the remainder of the swath, as shown in Figure 28, is recorded in a chain code notation tailored to fit the special requirements of the hybrid Vaster structure.

Given the presence of raster-formatted data within the Vaster data structure and with the remainder of the data locationally keyed to these raster-formatted positions, the spatial ordering and preservation of spatial relationships inherent in the raster structure is retained to a significant degree. In addition, the presence of both raster- and vector-formatted data in the Vaster structure offers retrieval and processing advantages for cartographic data base applications. Functionally, the Vaster format contains a two-level hierarchy. The raster portion of swaths may be used alone, without need to reference the more detailed vector file for many common processes or portions of processes, thereby speeding up performance considerably. A significant advantage of the use of this hybrid data structure for a very large data base is the ability to make use of only the raster file as a set of generalized data for quick browsing or sampling of the data base. This separation of use requires that the raster portion of the data base be stored as a separate file in order to maximize speed in accessing the raster data as well as for more efficient data retrieval based on spatial criteria.

Many queries could utilize only the coarser, raster structured portion of the data base where approximate solutions are desired, as may be the case in many centroid, area, perimeter or arc-length calculations. Most spatial relation queries such as point inclusion and relative position could also be answered from raster-formatted data, using raster oriented algorithms which are generally more efficient for this type of task because they ignore the bulk of the data. These algorithms would operate in the conventional manner, as described by Merrill (1973). The margin of error between approximate measures derived from the raster data alone is a function of both the width of the swath (i.e., the sampling frequency) and the sinuosity of the map lines themselves. If this margin of error is unacceptable in the case of spatial relation queries, only a small portion of the

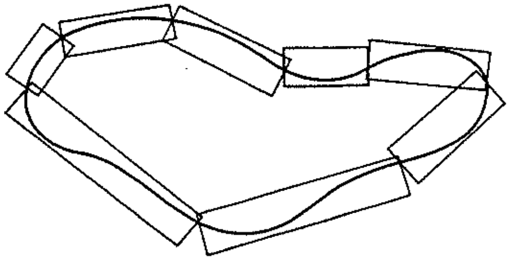


FIGURE 29. *The strip tree model.*

vector-formatted data need be referenced. For example, to determine containment of a point within an enclosed polygon where the y-coordinate of the given point falls between two scan-line records (that is, within a swath), then only the chain code boundaries of the particular polygon in question within the pertinent swath is converted to raster format so that the Jacobsen Plumbline algorithm can be performed using the same y value as the given point (Merrill, 1973).

There are other processes for which raster data organization will increase the efficiency of vector algorithms given the preservation of spatial ordering in the Vaster format. The two foremost examples are mosaicking and overlay of Vaster-coded data when these procedures are desired at full resolution. The approach to both of these processes is to first perform mosaicking or overlay on the raster-formatted portion of the data in each swath. This serves to 'align' the remaining chain-coded short line segments for easier vector-mode processing.

The Vaster structure may also be utilized as a full-detail, vector, chain-coded file. Any algorithms applicable to vector chain codes can be utilized in the normal manner after reconnecting the short chain segments of individual swaths by reference to the index (scan-line) records. Speed is thus maximized when a Vaster-formatted spatial data base is used for orientation and approximate information over large areas and more detailed analyses, based on chain-coded vectors, are reserved for relatively small areas.

The Vaster data format would seem to provide a 'best of both worlds' solution to the efficient storage and handling of spatial data. Large volumes of map data can be digitized via scanners, converted only part-way to vector format, and then be utilized by a wide variety of both raster-oriented and vector-oriented algorithms in the conventional manner. Conversion to either full raster or vector format would require much less time than conventional raster-to-vector and vector-to-raster procedures. Overlay with data stored in raster format, such as LANDSAT imagery is thus greatly facilitated.

The price paid for this, however, is the problem of determining the optimum data storage resolution. The problem has two related aspects. The first is a data sampling problem analogous to the grid cell size problem. How narrow must the swaths be to provide a satisfactory interval between raster-formatted records for the first-level data resolution? Each significant map line should be intersected at least once by a scan line, thus avoiding occurrences of map entities which are completely contained within the vector portion of a single swath. Closely spaced scan lines (i.e., narrow swaths relative to map line density and sinuosity), while avoiding such occurrences, cause multiple scan-line intersections of the same continuous map line, which can inflate data volumes.

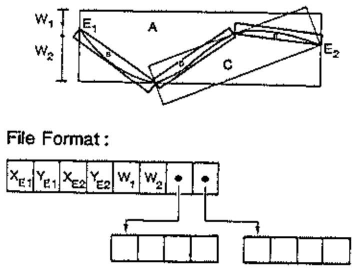


FIGURE 30. *Hierarchical structure of strip tree model.*

A second aspect of the problem arises from the hybrid nature of the Vaster structure. What is the raster-vector data volume ratio which would provide optimum overall performance for a given group of raster- and vector-oriented algorithms for a given task? It is the joint use of rasters and vectors which takes this question a level beyond that of a standard system optimization problem. The two aspects of this complex problem are also highly interrelated.

4.2.2 Strip Tree model

Strip trees, as developed by Ballard (1981), is a method for representing map vectors by means of a hierarchy of bounding rectangles (Figure 29). This is also a hybrid data model since the basic logical entity of the model is the cartographic line (i.e., vectors), but the lines themselves are not explicitly recorded. This representation is designed to allow such operations as union, intersection and length of curves to be performed efficiently.

The representation of curves consists of a binary tree structure where lower levels in the tree correspond to finer resolutions (Figure 30). The tree structure was derived from using Duda and Hart's method for digitizing lines and retaining all intermediate steps in the digitizing process (1973).

The idea of representing a cartographic line by a hierarchy of strips was first presented by Peucker. He was able to use this data model to find line intersection and point-in-polygon (1976). Burton used a similar but more general form where curves are divided into a hierarchy of bounding rectangles of a single orientation. These are then represented in a binary tree hierarchy (Burton, 1977).

As defined by Ballard, an individual strip is defined to consist of four elements, E1, E2, W1, W2, where E1 (location XE1 YE2) and E2 (location XE2 YE2) denotes the beginning and end of the strip, respectively, and W1 and W2 denote the right and left distances of the strip borders from the directed line segment. These definitions are depicted in Figure 30.

A sequence of nested strips between the endpoints of a line is derived by finding the smallest bounding rectangle between those two points. Next, a point is picked which touches one of the two sides of the rectangle and the process is repeated for each of the two sublists (Figure 30). This results in two subtrees which are sons of the root node. The process terminates when the lowest level strips have a width equal to, or less than, a given resolution. The data record for each strip thus contains the x and y coordinates for each of the endpoints, w1, w2, and the pointers to the two descendant strips.

The binary tree resulting from this process is called a strip tree. Other operations which can be performed on strip trees include testing the proximity of a point, displaying a curve at different resolutions, intersecting two trees, the union of two trees, point-in-polygon, and area union and intersection.

This model, thus, could be viewed as the vector counterpart to quadtrees, since each exploits a hierarchical structure based on squares or rectangles with similar operational advantages. The primary difference in this respect between quadtrees and strip trees is that in order to do many of these operations, such as overlay, the quadtrees must be spatially registered. This is not the case with strip trees. Strip trees can be arbitrarily translated and scaled since they can be defined in terms of points which are grid independent.

5.0 FUTURE DEVELOPMENTS IN SPATIAL DATA HANDLING

5.1 GLOBAL DATABASES

A large-area database which covers, for example, all of the U.S., or all of North America, can adjust for the curvature of the earth by conversion of the pertinent portion of the database to the appropriate map projection. Thus, a data file stored in lat-long coordinates can be converted to, say, an Albers Conic Equal-Area if the data for the U.S. plotted on a small-scale map are desired. Equal area projection coordinates are also necessary if large-area areal calculations are to be performed.

Vector-type data models, employing a spherical coordinate system such as latitude-longitude, pose no special problems on a global scale. The overwhelming proportion of geographic data being generated, however, is grid or raster based, such as LANDSAT imagery.

It is, unfortunately, topologically impossible to cover any spherical surface, such as the earth, with a square or rectangular grid. Although it is possible to convert stored coordinates for a portion of the database to an appropriate projection for analysis or map display, the real problem is to design a single, integrated global data model for representation of the database. The model should not have any spatial gaps or overlaps on the globe. For a tessellation-type model, it is assumed that a regular tessellation of an equilateral polyhedron is desired.

Given these general requirements, we are limited to five shapes. This fact was known to the ancient Greeks and its proof may be found in any textbook on solid geometry. These five shapes are called the platonic solids, having been discussed by Plato. They consist of 1 the regular tetrahedron, the four faces of which are equilateral triangles; 2 the regular hexahedron, or cubic; 3 the regular octahedron, the eight faces of which are equilateral triangles; 4 the regular dodecahedron, the twelve faces of which are regular pentagons; and 5 the regular icosahedron, the twenty faces of which are equilateral triangles (Fisher, 1944; Coxeter, 1973) (Figure 31).

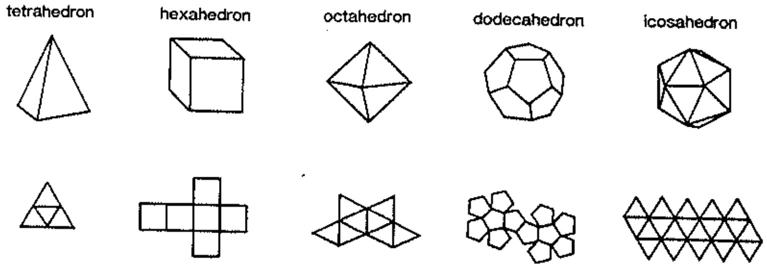


FIGURE 31. *The five Platonic solids.*

For purposes of a global data model, the dodecahedron is not a good choice because the pentagons cannot be recursively subdivided into other pentagons. The tetrahedron and hexahedron have faces which are too large in relation to the globe and would introduce significant areal and shape distortions. With a choice between the octahedron and icosahedron, the latter is the obvious choice since they both have triangular faces, but the individual flat faces of the icosahedron are smaller and would cause less distortion in representing portions of a spherical surface.

Using the 20 faces of the icosahedron, continued recursive subdivision into smaller triangles provides less and less distortion. Fuller and Sadao (1982) have published a projection based on this polyhedron, termed the 'dymaxion' projection. This projection is not widely used and is regarded by cartography professionals as a curiosity. The primary reason for this is that when laid out on a flat piece of paper, it presents a very distorted view of the earth because of the pattern of interruptions. Nevertheless, when viewed as a 'virtual map' in computer storage where visual perceptions are no longer of consequence, the icosahedron has a number of useful properties (Dutton, 1983; Fuller, 1983). These present intriguing possibilities for use as a general purpose global data model. A regular spherical tessellation retains all of the desirable properties of a planar tessellation including implicit spatial relationships; geographic location is implied by location in the database, variable scale and rapid search.

The major drawback of this model for a high-resolution global database is that the number of triangles at each successive subdivision would increase geometrically, producing a data volume explosion of unmanageable size. Upon closer inspection, however, it is noticed that the size of each triangle must also decrease geometrically with successive subdivisions. The result is that small triangles are achieved after a surprisingly few subdivisions. For example, after eleven subdivisions (i.e., level twelve of a triangular quadtree) each triangle has an edge length of only 17.2 miles. This yields a cumulative total of 1,417,176 faces (Dutton, 1983).

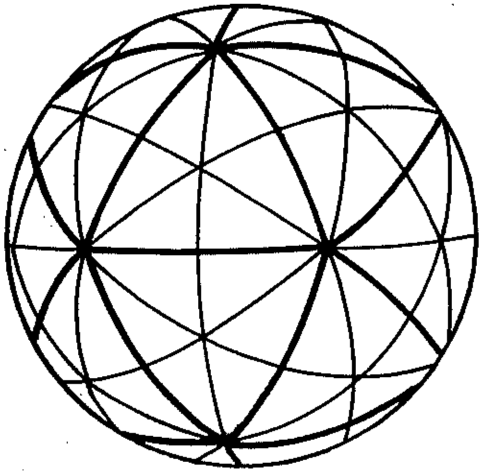


FIGURE 32. *Fifteen symmetrically arranged great circles on the sphere.*

The global triangular tessellation has other interesting properties as a global cartographic projection. When projected on a sphere, the original twenty triangles remain equal-area and equilateral. The edges of these triangles also form a set of great circle arcs. However, there is shape distortion from the original planar triangles created. A measure of the variations in linear scale can be calculated by comparing the projected lengths of the equal great-circle arc segments with the lengths of those segments before projection. Also, the vertices of the twenty triangles provide a measure of angular distortion or departure from conformality. Since, as a regular network, these triangles are evenly distributed over the surface of the globe, it allows a fair comparison to be made between different parts of the same projection (Fisher, 1944).

A further problem is that subdivision into smaller triangles on the sphere does not produce equal area triangles. Nevertheless, the methods for calculating the exact size of any individual triangle at any position and level in a recursive hierarchy, as well as the length of its sides and the angles of its vertices is well known from the field of spherical trigonometry. By being selective in further subdividing the sphere based on the 30 edges of the icosahedron, there are some unique possibilities. If six great circle arcs are drawn from the centers of each of the original twenty triangles, three passing through the mid point of each edge and three to each corner, the mesh is equally subdivided into 120 right-angled triangles (Figure 32). This complicated pattern would include, among others, a pattern of thirty equal quadrilaterals and a series of overlapping hexagons, in addition to the original twenty triangles. The possibilities of this as a compound tessellation are tantalizing as a best-of-all-worlds approach, but the operational geometry is complex.

5.2 SPACE-TIME DATA

By far, the major emphasis in the development of geographic data models has been on two-dimensional models. This was a result of many factors including: 1 demand; 2 the state-of-the-art in analytical techniques in a number of fields; 3 limitations in computing capacity; and 4 the state-of-the-art in data modeling techniques. A significant amount of work on multidimensional models for specialized applications, such as 3-D TIN models for terrain analysis, has also been done.

Only within the past few years has there been broad-based attention on the development and techniques for handling of time-series data. On the surface, it would seem that extending two-dimensional and three-dimensional models into the fourth dimension should be a straightforward process. This turns out not to be true in practice, however.

Using the example of spacecraft data, an individual instrument collects data at frequent time intervals, and often over the span of years. This type of time series data collection generates enormous data volumes. The problem encountered in this area derives from two factors. 1 Time is different in nature in comparison with the space dimensions. It is entropic and runs in only one direction. 2 Very little is known about the entities and processes portrayed, particularly for non-earth data. Inferences, therefore, cannot be made from 'similar' occurrences in other locations.

These two factors make compression via sampling of the time dimension without reducing or obscuring the information contained in the data virtually impossible. Since valid time series data have been virtually non-existent in many fields until recently, detailed analyses of how various phenomena and their interrelationships change over time could not be made. Compression can therefore not be accomplished by exploiting derived or previously known statistical dependencies that exist between separate time samples. Compression by discarding data which would not be of interest to the user is also difficult because of lack of experience with this type of data. A primary means of analysis, such as was the case for the *Voyager I* flyby, is therefore to graphically portray complete sequences of time-series data as 'movies' to be subjected to expert visual interpretation. Methods for digitally storing and analyzing space-time data are thus an area currently in need of further investigation.

5.3 SUMMARY AND CONCLUSIONS

As stated in the introduction of this paper, geographic databases currently in existence are experiencing severe problems of inefficiency, in terms of both compaction and speed of use, as well as rigidity and narrowness in the range of applications and data types which can be supported by a single database. These efficiency, versatility and integration problems can be traced in large part to the profound differences in the storage formats commonly used for spatial data handling. The basic problem, however, is a lack of understanding of the nature of spatial data, and a lack of a unified body of knowledge of the design and evaluation of spatial data structures.

This problem is particularly critical for many national-level governmental agencies worldwide, since they collect and utilize very large volumes of data of a wide range of types, from diverse sources, with varying accuracy, resolution and format characteristics as a basic and essential part of their function. These data sets must also frequently satisfy a wide range of scientific applications, the data needs for which are simultaneously vague and changing over time. Satisfying the unexpected request in this situation becomes the rule rather than the exception.

This paper has shown through an examination of current knowledge and spatial data models in a comprehensive framework, that major advances in the performance of spatial data models and geographic information systems are both possible and probable within the next few years which will allow us to meet these efficiency, versatility and integration needs.

The taxonomy presented here focused primarily on 2-dimensional spatial data models. Extension of these models to 3-dimensions is straightforward. However, as discussed in section 5.2, this does not carry through to the 4th dimension, time. This is the one area currently identified in spatial data models and computer spatial data handling where we have barely scratched the surface.

The taxonomy given in this paper has provided clarification of how varying data models are conceptually interrelated. Many of the gaps in the options available have been filled by very recent research, particularly in the areas of nested tessellations and hybrid data models, and by recent recognition of long-known principles that were developed in other disciplines. This taxonomy can thus provide a framework for future work in the systematic analysis and comparison of performance characteristics of different types of spatial data models.

This knowledge on performance characteristics can serve as guidelines for building geographic information systems in the future in a more predictable and systematic manner. Performance characteristics can also serve as a guide for direction in future refinements and developments in spatial data structures and algorithms by revealing deficiencies.

BIBLIOGRAPHY

ABEL, D.J. and J.L. SMITH. 1983. A data structure and algorithm based on a linear key for a rectangle retrieval problem, *Computer Vision, Graphics & Image Processing*, vol. 24, pp. 4–14.

AHUJA, N. and BRUCE SCHACTER. 1981. Image models, ACM *Computing Surveys*, vol. 13, pp. 373–397.

AHUJA, N. 1983. On approaches to polygonal decomposition for hierarchical image decomposition, *Computer Vision, Graphics & Image Processing*, vol. 24, pp. 200–214.

AURENHAMER, F. and H. EDELSBRUNNER. 1984. An optimal algorithm for constructing the weighted Voronoi diagram in the plane, *Pattern Recognition*, vol. 17, pp. 251–257.

BALLARD, D. 1981. Strip trees: a hierarchical representation for curves, *Communications of the* ACM, vol. 24, pp. 310–321.

BENGTSSON, B. and S. NORDBECK. 1964. Construction of isarithms and isarithmic maps by computers, *Nordisk Tidschrift for Informations-Behandling*, vol. 4, pp. 87–105.

BENTLEY, J.L. 1975. Multidimensional search trees used for associative searching, *Communications of the* ACM, vol. 18, pp. 509–517.

BOARD, C. 1967. Maps as models, in *Models in Geography*, ed. P. Haggett, pp. 671–725, Methuen & Co., Ltd., London.

BOOTS, B.N. 1979. Weighting Thiessen polygons, *Economic Geography*, pp. 248–259.

BOUILLÉ, F. 1978. Structuring cartographic data and spatial processing with the hypergraph-based data structure, *Proceedings of the First International Symposium on Topological Data Structures for Geographic Information Systems*, Harvard Univ., Cambridge.

BURTON, W. 1977. Representation of many-sided polygons and polygonal lines for rapid processing, *Communications of the* ACM, vol. 20, pp. 166–171.

CEDERBERG, R. 1979. Chain-link coding and segmentation for raster scan devices, *Computer Graphics & Image Processing*, vol. 10, pp. 224–234.

CHAKRAVARTY, I. 1981. A single-pass, chain generating algorithm for region boundaries, *Computer Graphics & Image Processing*, vol. 15, pp. 182–193.

CODD, E.F. 1970. A relational model of data for large shared data banks, *Communications of the* ACM, vol. 13, pp. 378–387.

——— 1981. Data models in database management, *Proceedings*, Workshop on data abstraction, databases and conceptual modelling.

COXETER, H.S.M. 1973. *Regular polytopes*, Dover Publications, Inc., New York.

DACEY, M. and D. MARBLE. 1965. Some comments on certain technical aspects of geographic information systems, Technical Report No. 2 of ONR Task No. 389-142, Office of Naval Research, Geography Branch.

DANGERMOND, J. 1982. A classification of software components commonly used in geographic information systems, *Proceedings, U.S.-Australia Australia Workshop on the Design and Implementation of Computer-Based Geographic Information Systems*, pp. 70–91, Honolulu, Hawaii.

DATE, C.J. 1983. *An introduction to database systems*, vol. 11, Addison-Wesley, Reading, Mass.

DRYSDALE, R. and D. LEE. 1978. Generalized Voronoi diagram in the plane, *Proceedings, 16th Annual Allerton Conference on Communications Control and Computers*, pp. 833–839.

DUDA, R.C. and P.E. HART. 1973. *Pattern classification and scene analysis*, Wiley-Interscience, New York.

DUTTON, J. 1983. Geodesic modelling of planetary relief, *Proceedings, Auto-Carto VI*, pp. 186–201, Ottawa.

FINKEL, R.A. and J.L. BENTLEY. 1974. Quad trees: a data structure for retrieval on composite keys, *Acta Informatica*, vol. 4, pp. 1–9.

FISHER, I. and O.M. MILLER. 1944. *World maps and globes*, Essential Books, New York.

FREEMAN, H. 1961. On the encoding of arbitrary geometric configurations, IRE *Transactions*, vol. 10, pp. 260–268.

——— 1961. Techniques for digital computer analyses of chain encoded arbitrary plane curves, *Proceedings, National Elec. Conf.*, pp. 421–432.

——— 1974. Computer processing of line-drawing images, *Computing Surveys*, vol. 6, pp. 57–97.

——— 1979. Analysis and manipulation of line-drawing data, *Proceedings of the* NATO *Advanced Study Institute on Map Data Processing*, Maratea, Italy.

FULLER, R.B. and E.J. APPLEWHITE. 1982. *Synergetics: explorations in the geometry of thinking*, MacMillan, New York.

FULLER, R.B. and S. SADAO. 1982. *Spaceship Earth edition of the Dymaxion sky-ocean map*, Buckminster Fuller Institute, Philadelphia.

GARGANTINI, I. 1982. An effective way to represent quadtrees, *Communications of the* ACM, vol. 25, pp. 905–910.

GIBSON, L. and C. LENZMEIER. 1981. Hierarchical pattern extraction system for hexagonally sampled images, report prepared for the Air Force Office of Scientific Research, Interactive Systems Corp.

GIBSON, L. and D. LUCAS. 1982. Vectorization of raster images using hierarchical methods, *Computer Graphics and Image Processing*, vol. 20, pp. 82–89.

GOLD, C.M. 1978. The practical generation and use of geographic triangular element data structures, *Proceedings of the First International Symposium on Topological Data Structures for Geographic Information Systems*, Harvard Univ., Cambridge.

GOLDBERG, M. et al. 1984. The topological grid structure: a data structure for Earth science modelling, *Proceedings, Pecora* IX *Symposium*, Sioux Falls, S.D.

HUNTER, G.M. 1978. Efficient computation and data structures for graphics, Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, N.J.

HUNTER, G.M. and K. STEIGLITZ. 1979a. Linear transformation of pictures represented by quadtrees, *Computer Graphics and Image Processing*, vol. 10, pp. 289–296.

——— 1979b. Operations on images using quad trees, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 145–153.

IGU COMMISSION ON GEOGRAPHICAL DATA SENSING AND PROCESSING, 1975. Information systems for Land Use Planning, report prepared for Argonne National Laboratory.

——— 1976. Technical Supporting Report D, U.S. Dept. of the Interior, Office of Land Use and Water Planning.

JACKINS, C.L. and S.L. TANIMOTO. 1980. Oct-trees and their use in representing three-dimensional objects, *Computer Graphics and Image Processing*, vol. 14, pp. 249–270.

——— 1983. Recursive decomposition of Euclidean space, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 533–539.

KLINGER, A. 1971. Patterns and search statistics, in *Optimizing Methods in Statistics*, J.S. Rustagi, ed., pp. 303–337, Academic Press, New York.

KLINGER, A. and C. DYER. 1976. Experiments on picture representation using regular decomposition, *Computer Graphics and Image Processing*, vol. 5, pp. 68–105.

KLINGER, A., K.S. FU, and T.L. KUNII. 1977. *Data structures, computer graphics, and pattern recognition*, Academic Press, New York.

KNUTH, D. 1969. *The art of computer programming, volume I: Fundamental algorithms*, Addison Wesley, London.

——— 1973. *The art of computer programming, volume III: Sorting and searching*, Addison-Wesley, Reading, Mass.

KOPEC, R.R. 1963. An alternative method for the construction of Thiessen polygons, *Professional Geographer*, vol. 15, pp. 24–26.

LEE, D.T. 1980. Two-dimensional Voronoi diagrams in the LP metric, *Journal of the* ACM, vol. 27, pp. 604–618.

LOHMAN, G.M. et al. 1983. Remotely-sensed geophysical databases: experience and implications for generalized DBMS, *Sigmod Record*, vol. 13, pp. 146–160.

MACKAY, J.R. 1953. The alternative choice in isopleth interpolation, *Professional Geographer*, vol. 5, pp. 2–4.

MANDELBROT, B.B. 1977. *Fractals: form, chance, and dimension*, W.H. Freeman and Company, San Francisco.

MARK, D. and J.P. LAUZON. The space efficiency of quadtrees: an empirical examination including the effects of two-dimensional run-encoding, *Computer Vision, Graphics and Image Processing*, forthcoming.

MARK, D.M. 1975. Computer analysis of topography: a comparison of terrain storage methods, *Geografiska Annaler*, vol. 57a, pp. 179–188.

——— 1978. Concepts of data structure for digital terrain models, *Proceedings of the* DTM *Symposium*, pp. 24–31, St. Louis.

——— 1979. Phenomenon-based data-structuring and digital terrain modelling, *Geo-Processing*, vol. 1, pp. 27–36.

MARTIN, J. 1975. *Computer data-base organization*, Prentice Hall, Inc., Englewood Cliffs.

MERRILL, R.D. 1973. Representation of contours and regions for efficient computer search, *Communications of the* ACM, vol. 16, pp. 69–82.

MORTON, G.M. 1966. A computer oriented geodetic data base; and a new technique in file sequencing, unpublished report.

NYERGES, T.L. 1980. *Modeling the structure of cartographic information for query processing*. Unpublished Ph.D. dissertation, Ohio State University.

PEANO, G. 1973. *Selected works*, Ed. H.C. Kennedy. Toronto: University of Toronto Press.

PEUCKER, T. and N. CHRISMAN. 1975. Cartographic data structures, *The American Cartographer*, vol. 2, pp. 55–69.

PEUCKER, T. 1976. A theory of the cartographic line, *International Yearbook for Cartography*, vol. 16, pp. 134–143.

——— 1978. Data structures for digital terrain models: discussion and comparison, *Proceedings of the First International Symposium on Topological Data Structures for Geographic Information Systems*, Harvard Univ., Cambridge.

PEUQUET, D. and A.R. BOYLE. *Raster scanning, processing and plotting of cartographic documents*, forthcoming.

PEUQUET, D. 1979. Raster processing: an alternative approach to automated cartographic data handling, *American Cartographer*, vol. 6, pp. 129–139.

——— 1981a. Cartographic data, part I: the raster-to-vector process, *Cartographica*, vol. 18, pp. 34–48.

——— 1981b. An examination of techniques for reformatting digital cartographic data, part II: the vector-to-raster process, *Cartographica*, vol. 18, pp. 21–33.

——— 1983. A hybrid structure for the storage and manipulation of very large spatial data sets, *Computer Vision, Graphics and Image Processing*, vol. 24, pp. 14–27.

REDDY, D.R. and S. RUBIN. 1978. Representation of three-dimensional objects, Report #CMU-CS-78-113, Computer Science Department, Carnegie-Mellon.

RHYNSBURGER, D. 1973. Delineation of Thiessen polygons, *Geographical Analysis*, vol. 5, pp. 133–144.

SAMET, H. 1984. *The quadtree and related hierarchical data structures*, ACM *Computing Surveys*, vol. 6, no. 2.

SCHOLTEN, D.K. and S.G. WILSON. 1983. Chain coding with a hexagonal lattice, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 526–533.

SENKO, M.E. et al. 1973. Data structures and accessing in data base systems. IBM *Systems Journal*, vol. 12, pp. 30–93.

SENKO, M.E. 1976. DIAM 11 and levels of abstraction. *Proceedings*, Conference on Data: Abstraction, Definition and Structure, vol. 8 pp. 121–140.

SHAMOS, M. 1978. *Computational geometry*, Ph.D. Dissertation, Yale University.

STEVENS, R.J., A.F. LEHAR, and F.H. PRESTON. 1983. Manipulation and presentation of multidimensional image data using the Peano scan, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 520–526.

TANIMOTO, S. and T. PAVLIDIS. 1975. A hierarchical data structure for picture processing, *Computer Graphics and Image Processing*, vol. 4, pp. 104–119.

THIESSEN, A.H. 1911. Precipitation averages for large areas, *Monthly Weather Review*, vol. 39, pp. 1082–1084.

TOBLER, W. 1959. Automation and cartography, *Geographical Review*, vol. 49, pp. 526–534.

——— 1978. Data structures for cartographic analysis and display, *presented at the Eleventh Annual Symposium on the Interface: Computer Science and Statistics*, Raleigh, N.C.

TOMLINSON, R.F. 1973. A technical description of the Canada Geographic Information System, unpublished report.

TOMPA, F.W. 1977. Data structure design, in *Data structures, computer graphics and pattern recognition*, T.L. Kuni, ed., Academic Press, New York, pp. 3–30.

TROPF, H. and H. HERZOG. 1981. Multidimensional range search in dynamically balanced trees, *Angewandte Informatik*, pp. 71–77.

US DEPT. OF COMMERCE, BUREAU OF THE CENSUS. 1970. The DIME geocoding system, in *Report No. 4, Census Use Study*.

ULLMAN, J. *Principles of database systems*. 1983. Computer Science Press, Rockville, Md.

WHITE, M. N-trees: large ordered indexes for multi-dimensional space, *American Cartographer*, forthcoming.

RESUME On examine les principaux types de modèles de données spatiales actuels, et on les place dans un cadre global. Ce cadre est utilisé pour éclaircir comment ces divers modèles spatiaux, de même que leurs avantages et désavantages, sont interreliés. Le cadre laisse aussi voir comment ces demandes contradictoires peuvent être équilibrées d'une façon plus systématique et prévisible pour des applications pratiques, et révèle les directions que doit prendre la recherche future.

ZUSAMMENFASSUNG Dieser Artikel untersucht die Hauptarten der zur Zeit bekannten räumlichen Datenmodelle und stellt sie in einen umfassenden Rahmen. Diesen Rahmen verwendet der Verfasser, um die gegenseitigen Beziehungen der Datenmodelle sowie deren Vor = und Nachteile aufzuzeigen. Ausserdem bietet der Rahmen Hinweise, wie man die oft widerstreitenden Anforderungen der Modelle systematischer und bestimmbarer ausgleichen kann, und gibt Anleitungen für weitere Forschungen.

RESUMEN. Este trabajo exmania los principales tipos de modelos de datos especiales actualmente conocidos y coloca estos modelos en un marco comprensivo. Se utiliza este marco para clarificar como varios modelos de datos, así como sus ventajas y desventajas inherentes, están interrelacionados. También da una idea en cuanto a como estas demandas conflictivas pueden balancearse de una manera más sistemática y pronosticable para aplicaciones prácticas y ofrece orientación para la investigación futura requerida.