#1128

Spatial Concepts, Geometric Data Models and Data Structures[1]

Andrew U. Frank
Department of Surveying Engineering and
National Center for Geographic Information and Analysis
University of Maine
Orono ME 04469

(207)-581 2174
email: FRANK@MECAN1.bitnet

Abstract

There seems to be some uncertainty in the GIS literature regarding the use of the words data model and data structure. There is a clear understanding of these notions in the database literature and it is possible to define analogous terms for GIS: geometric data model and geometric data structure.
Geometric data model is used to describe a formalized abstract set of spatial object classes and the operations performed on them.
Geometric data structure is then the specific implementation of a geometric data model, which fixes the storage structure, utilization and performance.
Humans organize their spatial perceptions using concepts that can be defined as **spatial concepts** to denote an informal or not directly implementable conceptual structure used to understand space.
Examples are given to clarify the theoretical discussion.

1. Introduction

Discussions of data structures to model geometry for geographic information systems (GIS) have progressed considerably over the last 15 years. The key issue is to model geometric concepts describing reality using a computer system. Although this does not seem difficult, research and development efforts of recent years have often contributed more to an appreciation of the problem than to a final solution.
Initially, the problem was considered one of optimal data structures on a very low level, close to the organization and operations of the basic computer hardware. Discussion of this topic can be found in (Dutton 1979). Research during this time was concerned with the computer aided treatment of cartographic data and the industry produced computer assisted map maintenance systems. At the same time, there were papers discussing the analytical capabilities that a geographic information system could offer to geography and other geosciences. These functions appeared to be extremely attractive, but research indicated that models had to contain more than just the cartographic data.

---

[1]This paper represents part of Research Initiative #2, "Languages of Spatial Relations", of the National Center for Geographic Information and Analysis, supported by a grant from the National Science Foundation (SES-88-10917). Support by NSF is gratefully acknowledged.

Frank, Andrew U. "Spatial Concepts, Geometric Data Models and Geometric Data Structures." *Computers & Geosciences* 18, no. 4 (1992): 409-17.

Data structures to represent geometric data were also needed in CAD/CAM (computer aided design/computer aided manufacturing) systems. These systems were initially developed to facilitate the production of paper drawings (CAD) but with the promise of extending further into the design and manufacturing process. Similarly as in geographic information systems, the limitations of representing geometric concepts with the tools of traditional drawings became apparent.

Understanding the limitations of computer assisted map maintenance systems pointed the way to data structures which represent geometry, not the map image of geometric phenomena. In (Frank 1984) it was argued that there should be a differentiation between systems that deal with data directly representing some geometric reality and systems that deal with map representations. Only the former can support sophisticated geometric analytical functions, whereas the latter help human users to produce maps that can be analyzed by skilled users.

The discussion of geometric data structures often included treatments of the conceptual bases and the theoretical foundations but then detailed the implementation. For geographic information systems, two principal standard structures were established: vector and raster methods. Peuquet (1983) even proposed a compromise (vaster) idea. A very extensive literature for efficient implementation of raster structures using a quadtree data structure has been presented in (Samet 1989a; Samet 1989b).

Efforts to establish a theoretical base for geometric data structures came from different quarters. A landmark work (Corbett 1979) stressed the importance of topology as a basic mathematical concept for organizing geometric data. This paper, unfortunately not published in a widely circulating journal, is otherwise typical for its time: it contains extensive discussion of implementation at the hardware/assembly language level, which somewhat obscures its deep theoretical contribution. In (Frank 1983a) a graph theory based approach was found to be lacking. Peuquet (1988) used image processing concepts, and (Chan and White 1987) traced the origin of the 'map algebra' concept back to traditional methods used by urban planners.

In this context a number of issues relating to terminology arise. In the past, these issues have been the cause of some confusion and an attempt to resolve them is made here. In the database literature similar problems have been dealt with for quite a number of years and terminology is well established. Geographic information system should not invent new terminology, but use and extend by analogy, established information system and database terminology.

This paper will concentrate on the three notions of **spatial concepts, geometric data models and geometric data structures**. It will be shown that these are three different concepts which need to be separated. Each of the topics will be described in turn and some examples will be presented. The discussion will conclude with an overview of alternative viewpoints and the problems that can be resolved adopting the viewpoint purported here.

2. Data models and data structures

One of the reasons for building generalized database management systems was the observation that it was possible to program the low level **data structures** and the related access mechanism only once and make these generally useful methods available to many different applications. Work started with concepts like index sequential access methods (ISAM) and general purpose sorting and merging routines and progressed to hashing and tree structures. A complete and authoritative survey of all these data structures is given in (Knuth 1973) for most ordinary (i.e. non-spatial) problems.

At the level of organization of data, early database management systems can be seen as generalized packages permitting the use of sort and search methods in an integrated package. Anyone who has tried to use a package of subroutines - and code, for these same functions are readily available today as packages of reusable routines - is well aware that the adaptation of such routines to a specific task is no minor feat. In order to describe the functionality of the database management system without including all details of the data structure etc. a simplified model of the data storage system was created. Most of the details of the specific data structure are implicit in this model. Indeed it was explicitly demanded that the **data model** should be generic and independent from the implementation or the specific hardware configuration in order to increase portability of an application and to insure hardware independence of the application programs (Codd 1982).

Much of the early database management system discussion centers around the selection of the appropriate abstraction and **data models**, with the clear understanding that there is a trade off between higher level of abstraction and more automatic solutions vs. lower levels of abstraction, more adaptability and thus (most often) higher performance. Different companies offered database management systems with different interfaces, with very significant differences in the ease of use or level of knowledge necessary to understand and use the system (CODASYL 1971).

Data models thus evolved from an effort to find the common functionality and provide an abstract model of typical implementations. In 1970 E.F. Codd defined a **data model** from a top-down position. He defined a conceptually simple data structure with an appropriate set of operations, the relational data model (Codd 1970). The stress on a data model thus focussed on a conceptually simple construction - which can be implemented in more than one way - and which explains the database management system behavior. From a database administrator's point of view, the data model defines the interface from the database management system. It can thus be said, that 'the data model defines the tools available to structure the data' (Zehnder 1981) which will be stored in the database. This is essentially the same as the definition of the data model as 'a set of guidelines for the representation of the logical organization of the data in a database .. (Consisting) of named logical units of data and the relationships between them' (Tsichiritzis and Lochovsky 1977).

From a modern point of view, it should be stressed that a data model is a set of objects with the appropriate operations and integrity rules formally defined (Ullman 1988; Date 1986). This is essentially the definition for an algebra and it is therefore appropriate to speak of a relational algebra. The specific object types are selected such that they can be used to explain or define the structure in data, and there is often a specific data description language defined. This data description language is then used to describe the specific data used in an application or organization and stored in a database. This is the so called **database schema**. The concepts are selected so that they can be implemented.

The database community uses the notions of a **'data structure'** which is a generic or specific set of methods or programs to access data, which is stored in a specific way, and **'data model'**, which is a generic, highly abstract set of concepts with which a database administrator can describe the data and their relationships. We propose to use the same concepts for geographic information system, arguing that geographic information system face essentially the same problems and are constructed similarly to other information systems.

A database managment system based on a particular data model is constructed by selecting a data structure which offers the operation the data model describes. Mathematically speaking, one maps the abstract operations of the data model to the implemented operations (the subroutines) of the data structure. This mapping must preserve the intended behaviour of the operations, it must therefore be a homomorphism.

3. Data models in spatial data: two examples

In geographic information system research there have been numerous discussions of data structures which could be used to represent spatial data and provide a useful set of operations; in a recent set of books (Samet 1989a; Samet 1989b) a large number of such structures are surveyed. There is also a need for more abstract concepts to describe geometric data and the appropriate operations, which are independent from a specific implementation (Goodchild 1990; Frank and Mark 1990).

To clarify the notions of data model and data structure as applied to spatial problems, two major examples will be presented, namely the so-called raster and topological (vector) data model and their underlying data structures. Goodchild (1990) shows that both these structures can be seen as different discretizations from the same spatial concept.

3.1. Raster data model

This popular data model is based on a regular raster which divides space in regular shaped and sized pieces. For each of these pieces one then records attribute values, either as averages or the values at some specific points. There are a (small number) of variants in the raster data model, as we may use any of a number of regular tessellation to subdivide space (Diaz and Bell 1986).
The typical operations on the raster data model combine the data from one raster cell (using the values for different properties) to compute a new data value for the same cell. This is a form of spatial overlay, which compares well with the practice used by planners (Chan and White 1987).

The use of a regular square raster to represent geometric values is a geometric data model, for which we can define an appropriate set of operations independent of the specifics of an implementation - as was first done in 'map algebra' by Dana Tomlin (1983;1989).

There are several methods to implement this geometric data model with its operations; from the obvious use of a FORTRAN ARRAY, through run length encoding, and quadtrees, with their specific variants of implementation, being among the most effective ones (Samet 1988)(Samet 1989a).

3.2. Topological data model

Another frequently used data model is based on a subdivision of space into irregularly shaped regions (often called cells) with their boundaries, formed by lines called arcs or segments which link points (called nodes). This model is based on mathematical topology (Alexandroff 1961) and includes operations to find the boundary of a given object etc. For geographic information systems use one needs further an operation to overlay one partition with another one and to determine the intersection areas. Such an operation obviously uses metric properties to calculate the points of intersection between boundaries etc. Thus, the data model is not purely topological.

A standard implementation uses records for nodes (with their position expressed as coordinate pair), records for areas with their values for the interesting properties, and records for arcs, which contains links to the start and end nodes for each arc and links to the area to the left and the right of the arc [Figure 1]. There are other implementation concepts that provide the same functionality (e.g. TIGRIS (Herring 1990) or the geo-relational algebra (Gueting 1988)).

Nodes (node-id, x, y)
Areas (area-id, property-value1, property-value2,...)
Arcs (arc-id, id of start-node, id of end-node, id of left-area, id of right-area)

Figure 1 Relational schema to implement the topological data model

In principle, results from operations on the same data and in the same data model but represented by different implemented data structures, should be the same. It was a major complaint when a federal agency tested geographic information systems which implemented a topological data model and found that the results from operations executed on one or the other yielded substantially different results.

4. Spatial concepts

The data models of the database management systems, available as data description language are useful in representing a specific perception of the world, described as a model of reality, populated with the values as available as data description language in the data sets. The way to model reality was often assumed as given, as these applications deal with artifacts (e.g. bank accounts, insurance policies, stock in warehouse) which were defined in an operative manner through the business practice. When software started to model real systems, i.e. a system which had an observable real counterpart, software engineers realized that there was an additional problem of how humans conceptualize reality. This is not a problem in most administrative applications as the business practice, rules and regulations define how things ought to be understood.

The problem is especially important with geographic information systems but not substantially different from say, building a knowledge base for some other non-trivial field, and closely related to the efforts to formalize the everyday world in 'naive physics'. We observed that humans seem to use several different methods to conceptualize space (Mark, Frank et al. 1989): we seem to use an essentially Euclidean geometry when we reason about the spatial arrangements on our table or other small areas, use a network-topology view when we plan a trip or navigate a car, etc. It is not that reality changes but the concepts utilized to structure our perception of the situation may vary (Neisser 1976; Lakoff 1987). In order to cope with the complexity of a real situation we have to abstract from details and concentrate on the aspects that are important for the task at hand (Mark 1989; Frank and Mark 1990).

The concepts used to understand space are often based on notions which cannot directly be implemented, either for lack of formal definition or for lack of discretization.

The imaging schemata (Lakoff and Johnson 1980), which are basic for spatial cognition, and include such fundamental spatial relations as inside, across etc. are explained in linguistic terms (Herskovits 1987) but not formally defined such that they could be implemented. They include distance expressions like 'near' and 'far' and expressions for directions between extended objects. For formal treatment, often an infinitely dense collection of points, as in point set topology or in euclidian geometry is assumed. Goodchild (1990) proposes a 'geographic reality' based on points and values for properties of interest at these points:

$\{x,y,z_1,.Z_n\}$.

This concept of a geographic reality can only be implemented, i.e. represented in a finite machine, after discretization Goodchild (1990) discusses extensively the different alternatives of discretization which lead to different data models in our terminology. Implementations can only deal with explicit representations for a finite number of objects, thus a discretization is necessary to reach an implementable data model. This discretization however, introduces its own noticeable artifacts.

We therefore reach the following intermediate conclusion, namely to differentiate between three notions at the implementation; logical; and conceptual level [Figure 2]:

Data structures (specifically geometric and spatial data structures):
Detailed and low level descriptions of storage structures (traditional data structures) and the pertinent operations, with details of how the desired effects are achieved. They will not only provide a specific function (i.e. fulfill the conditions of an operation) but also are fixed in terms of performance, storage utilization etc. - They are a specific solution for a generic problem.

Data models (specifically geometric data models):
A comprehensive set of conceptual tools to be used to structure data. They are defined formally and are constructed such that they can be implemented.

Concepts (specifically spatial concepts and geometry):
Ideas, notions and relations between them which are used by humans to organize and structure their perception of reality. They differ depending on the task at hand, the circumstances and the experience of the persons.
They are either
- Formally defined but cannot be implemented, due to fundamental restriction of computer systems (e.g. limitations of finite machines).
- Informal, i.e. not formally defined or not (currently) definable

Figure 2- Definitions of geometric data structure, geometric data model and spatial concepts

5. Examples of spatial concepts

In the past (Frank 1987) we have attempted in theoretical studies, the data that describe the non geometric properties. It is sufficient to abstract all attribute data to a vector of values of unspecified type, and no further interactions between specific operations on this vector and the spatial data need to be considered (geometric and non-geometric data in (Gueting 1988)). This provides a base level description of spatial data.

If we structure the data in entities, there may be some additional structure between the entities (e.g. sets of all parcels belonging to a person, an ordered list of all schools in a district according to their capacity etc.). These are non-spatial aspects and have to be dealt with with the regular tools of the (non-geometric) data model.

We will see that it is sometimes useful to base a geometric data model on a generic one (e.g. it has been attempted to model a cell based geometric data model in a database schema using the relational data model (Gueting 1988) similar to figure 1 above) and to map geometric operations to operations on the generic (non-geometric) data model.

A tentative set of spatial concepts are discussed in the following subsections. This list is not yet complete, and it is not even clear if a complete list is possible. There are other important spatial concepts, which are not included for various reasons, chief among them is a lack of clear understanding. A traditional view is to differentiate between an entity based view - space is constructed from objects that fill space - and a space oriented view, where each point in space has some properties. This view is philosophically well established - it can be considered to go back to Kant on one hand and to Descartes on the other. This is a very important, theoretical as well as practical differentiation, which leads to a number of different concepts and differences in the operations applicable.

Some of these spatial concepts lead to very similar geometric data models, usually by discretization. For example, going from an infinite point set to a discrete one or representing node positon with finite precision coordinates. It must, however, be noted that these discretization steps include quite noticeable changes and fundamental properties expected from geometric operations are lost. For example, geometric constructions using finite precision coordinate values not invariant under translation and rotation and strange artefacts surprise the unsuspicious user. These differences justify the use of a separate terminology.

5.1. Sets of points

Space is thought of as a collection of an infinite number of dimensionless points which form a continuum. Each point is identified by a coordinate value (mathematically this is equal to R x R for 2 dimensional space) and this model assumes that the space is continuous and that the distribution of points is dense overall. For each point - at least theoretically - exists a vector of attribute values that describe its properties. This is essentially the spatial concept that Goodchild mentions as 'geographic reality' (Goodchild 1990).

5.2. Thematic layers, surfaces

An attribute associated with space can be thought of as a continuous surface (with a single value of the attribute per point in space). This concept is used primarily for the topographic surface of the world but can be applied to other data. We may assume or not that the surface is smooth and continuously differentiable, or the values change at some boundaries abruptly.

5.3. Euclidean geometry

Euclidean geometry is an entity oriented spatial concept. The object it deals with are points and infinite lines, and the operations on them are explained by a set of axioms. There exists a mapping to coordinate space, with algebraic expression substituting for the euclidean constructions with ruler and compass. Each point is represented by a pair of real numbers and formulae that correspond to geometric operations are given. The basic foundations of this model is thus very similar to the point set model, but euclidean geometry structures space into discrete entities.

5.4. Partitions

A division of space in areas, such that all the areas sum up to the whole and no two overlap (i.e. they are pairwise disjoint) is often used. Subdivision of land into ownership parcels is thought of in this way, but also soils classifications are constructed following this concept. Mathematically, such a construction is known as partition. Practically, we find partitions that are constructed based on attribute values, i.e. the (connected) set of all points with a given attribute value (or a value in an interval, or set of values) and this leads to disjoint areas. These partitions are called 'categorical coverages' (Beard 1988). On the other hand, one often uses choropleth maps, which are partitions which were previously constructed, e.g. following political boundaries, for reporting census and similar statistical values (Robinson, Sale et al. 1984).

5.5. Delimited spatial entities

In lieu of partitions, one may just define spatial units, each with its boundaries, without enforcing that they be disjoint (i.e. without 'planar enforcement' (Goodchild 1990)). This concept is more of importance for conceptual reasons than for actual data collection, where the demand for completeness of data collection (one of the attributes of data quality (Robinson and Frank 1984)) forces automatically a partition concept.

5.6. Cell topology

Cell topology is another, mathematically based concept, related to the continuous space concept. In cell topology, we deal with cells, of dimension 0 (points, so called 0-cells), of dimension 1 (arcs, so called 1-cells), dimension 2 (areas, so called 2-cells) etc. We are primarily concerned with relations between these objects, the boundary and co-boundary relations: an arc bounds an area, an area is bounded (co-bounds) by an arc; the same for arcs, which are bounded by points. In pure topology, the exact spatial location of nodes and arcs is not important, solely the spatial neighborhood is relevant. Thus configurations may be changed, as long as no cutting, hole puncturing etc. occurs.

5.7. Graphs

Graphs are built from two sets of objects, nodes and arcs and the connections between them, called adjacency. Variants of graphs have 'directed arcs'. Graphs need not be planar (i.e. arcs may cross without being connected). There is a substantial set of algorithms known to compute properties of graphs. Graphs seem to be a good approximation to the concepts used for navigation with cars (where we have to follow roads, which form a graph) and other transportation problems, where a network of possible connections is given.

A variant of great practical importance is the network, where individual points on the arcs can be addressed (for example by distance from one of the nodes, milepost in (Goodchild 1990)).

5.8. Cognitive spaces

It is - so far - not clear what are the exact properties of the cognitive concepts people use to deal with space. Observing problems with extending concepts gathered from 'small scale spaces' to other situations, Zubin proposed tentatively a set of spaces (Mark, Frank et al. 1989), which reach from a more euclidian view to a more graph oriented one:

5.8.1. Omniperspective

The small space one can perceive , where the minds eye sees the object (e.g. a cup on a table) from all sides, even if only one side is actually visible.

5.8.2. Monoperspective

The case where a view of a space is collected from various glances and the connected view of space is constructed in the mind (e.g a room).

5.8.3. Scene

Single perspective, where one sees only one side of an object and cannot infer its other sides (e.g. the perception of a building from the street curb).

5.8.4. Territory

The navigational concept, where one forms a concept of space by combining various views and experiences from interaction with the space (e.g. a town).

5.9. Imaging schema

Johnson provides a clear statement of how an image-schemata-based model of cognition would operate:

> "... Much of the structure, value, and purposiveness we take for granted as built into our world consists chiefly of interwoven and superimposed schemata... *My chief point has been to show that these image schemata are pervasive, well-defined, and full of*

sufficient internal structure to constrain our understanding and reasoning. To give some idea of the extent of the image-schematic structuring of our understanding (...), consider the following partial list of schemata, which includes those previously discussed:

CONTAINER	BALANCE	COMPULSION
BLOCKAGE	COUNTERFORCE	RESTRAINT REMOVAL
ENABLEMENT	ATTRACTION	MASS-COUNT
PATH	LINK	CENTER-PERIPHERY
CYCLE	NEAR-FAR	SCALE
PART-WHOLE	MERGING	SPLITTING
FULL-EMPTY	MATCHING	SUPERIMPOSITION
ITERATION	CONTACT	PROCESS
SURFACE	OBJECT	COLLECTION

This brief list is highly selective, but it includes what I take to be most of the important image-schemata. If one understands 'schema' more loosely than I do, it might be possible to extend this list at length." (Johnson 1987, p. 126).

6. Geometric data models

The spatial concepts are typically not directly implementable, because they are assuming an infinite set of points (or another form of the same continuum assumption) and must be discretized. Discretization as the major modelling step is commonplace in geography (Goodchild 1990), but it is often just thought of as sampling and averaging over regular raster cells. Another limitation of spatial concepts is that some of them are not formalized, but just loosely described in terms of cognitive processes and experiments.

A geometric data model must have a well defined set of objects and operations on these objects. This fulfills the 'formal definition' requirement. The set of object (instances) must be finite, in order for the model to be implementable on a finite computer system. The behavior of the model is stated in terms of the effects of the defined (change) operations, which are observable with the given (observe) operations (Guttag, Horowitz et al. 1978). It is generally possible to map these geometric data models as specific application schemata to the traditional data models (excluding a discussion of performance aspects).

To illustrate, here follows a short list of geometric data models and their characteristics as found in a geographic information system.

6.1. Regular tessellations of space (raster)

We can model the continuous space by a finite set of small, regular shaped areas that tessellate it. This is a simple and useful method to discretize space, either by regular sampling, which determines the value for a specific location, or by averaging over the area involved.

6.2. Point sets using interpolation

We can record the value of an attribute at specific points, either regularly spaced on a grid or irregularly distributed and then provide an interpolation method which determines value for all intermediate points. There is a large number of variations on this theme, depending what arrangement of points is permitted and what interpolation methods are assumed.

6.3. Spaghetti

Spatial concepts may represented by simple lines - usually this model is connected to a cartographic modelization, which represents reality as a map and the data model is then used to represent the map (and thus indirectly reality). The lines itself may be modelled by a sequence of points, thought as connected by straight lines, or more sophisticated interpolation methods may be selected.

6.4 Graph

The graph concept can immediately be translated in a data model. In order to simplify implementation, restrictions are often imposed, which may include planarity of the graph.

The concept can also be extended:

- the edges are directed,
- locations on the edges are possible without introducing new nodes,
- connections on the nodes are not all equal (i.e. there is an internal graph in the nodes, which need not be planar - so called turn tables).

The connections between the arcs can be thought of as straight or may have detailed and determined form (again more or less restricted, depending on the implementation).

6.5. Topological data model

This model includes the topological concepts as well the partition concepts, as it appears difficult to implement a partition structure without the use of topological relations. The model is often restricted with limitations on

- The form of the edges (often just straight, or approximated by arc of circle or splines),
- The number of nodes per cell,
- The permission to create islands in cells or not.

Restricting the model to form a cell complex eliminates isolated nodes and edges which do not separate areas. The definition of operations on cells becomes much simpler if we demand that the cells form triangles forming simplicial complexes (in lieu of the more general cell complex) (Frank and Kuhn 1986)

6.6. Future research

We see that the data models, even if they can be reduced to a small number of typical ones, differ between implementations, because details of the implementation are allowed to 'show' at the conceptual user interface. This is usually justified by 'better' performance. However, these small differences are costly, as they hinder transfer of data between systems and generally communication between systems and their users. The proposed geographic data exchange standard (Moellering 1987) informally defines a number of concepts which can be used to form a geometric data model (i.e. terms like node, arc, polyline), but the exact meaning of these terms cannot be given without the framework of a formal, algebraic definition using operations.

It is an attractive and important research plan to define these geometric data models formally, i.e. as algebraic specifications (Goguen 1989). From such a definition it can be shown how the concepts relate to each other. It is demanded that we define mappings between these algebras, i.e. morphism which map objects and operations (see (Herring 1990; Mark, Frank et al. 1989)). To a certain extent, Goodchild (1990) attempts to show how objects of one model can be deduced from another one - implicitly proposing the point set concept as general.

7. Geometric data structures

There exists a large number of data structures, defined in more or less detail, to implement the geometric data model. Indeed, in the past it often seemed that one found first a geometric data structure which then implicitly defined a geometric data model. The geometric data model however should be the abstract view of the geometric data structure, not the other way round.

In order to see the difference between data model and data structure, one can simply observe that:
- **Data structure** is concerned with performance, storage utilization and other implementation details,
- The **data model** is concerned with function.
The term 'database schema' is used for the description of a particular model of reality (expressed in the tools provided by a data model).

If we formally describe the geometric data model, it is possible to test to what degree a data structure implements a model. In principle, one should not need to know the implementation details, and one implementation should be exchangeable for another one. The data structure should export exactly the operations defined in the model.

In the following we will only list a few of the major data structure, without details as there exist an enormous amount of variants for each of them.

7.1. Raster data structures

These implement the regular tessellation models. Implementation can be as straight array data structure, methods applicable to sparse array may work but best results are generally attained with methods to exploit spatial autocorrelation. The best known methods are run length encoding and hierarchical storage schemes, known as quadtrees (Samet 1989b). Holroyd (1990) discusses the problems of compression methods extensively.

7.2. Point sets

Data structures to store individual points can use either a tabular structure (and possibly some indexing methods for access) or exploit regularity in the distribution of the points in space, such that the location of a point can be inferred from the identifier (which often is directly mapped to a storage location and only implicitly represented).

Implementation of interpolation methods differs widely and there is extensive literature on different interpolation methods and how they are best carried out. The choice evidently depends also on the field of application, as some methods are better able to deal with certain special situations.

7.3 Topological data structures

The basics of implementing a topological data structure are well understood, but there are considerable differences between them. They differ in the exact data model used and in the details of the implementation. There is considerable literature on the subject in CAD/CAM, but a definite text is lacking.

8. Geometric data structures used for indexing

In most applications that store spatial data, access to the data is not only based on identifiers (e.g. parcel numbers, names of towns etc.) but also on spatial location. One needs to answer questions such as 'what is at location x,y?' Or 'find all objects inside *a window*'. The data model for this problem consists of data objects for which a spatial location and extent is defined in a coordinate system, and access operations retrieve all objects within a *window ;* or finds the closest neighbor object to a given object (Frank 1981; Frank 1983b; Frank and Barrera 1989).

A large number of geometric data structures were developed specifically for this multidimensional indexing purpose and a number of the data structures included above can be used as well. Access to multidimensional point data is mostly solved but there further research is necessary in optimal access methods to extended spatial objects. In (Buchmann, Guenther, et al. 1990) an updated overview of this interesting field can be found; it is not the primary concern of this article, because the indexing structure as such does not participate in the modelling of reality. It contributes a performance gain over an operation which could be in principle at least, executed without use of the indexing structure. It is possible to find all objects within a *window* just by sequential inspection of all stored geometric data objects. This is clearly impractical for most larger data collections, but this is only a performance issue not a modelling one. From a practical point of view, it was found that a geographic information system should use a spatial indexing structure, but results from comparison of different data structures used to this purpose indicate that the performance differences between them are minor and not of great practical concern.

9. Conclusions

This paper began with an examination of the use of the terms 'data model' and 'data structures' in the computer science oriented database literature. 'Data model' means a set of conceptual tools to describe the logical or conceptual structure of the data, whereas 'data structure' is used to describe a specific implementation of a data model. A data model describes on the abstract level objects and their behavior, but only the data structures fixes performance aspects like storage utilization and response time.

It was found that similar concepts apply to modelling the geometric aspects in a geographic information system and it was proposed that the term **'geometric data model'** should be used to describe an abstract view of geometry and geometric properties of objects. It is recommended that it is formalized using an object-oriented viewpoint as an algebraic structure with a set of objects, operations to construct, change and observe these objects and axioms (rules) which explain the result of the operations in terms of other operations. Geometric data models must be formally defined and it must be possible to implement them on a current computer system.

Geometric data structures are then specific implementations which provide the operation demanded in a geometric data model, using specific storage structures and algorithms. Data structures exhibit specific performance properties, storage utilization and speed of operations being the most important ones. They are optimized for certain cases and yet may not be suitable for other applications.

Geographic information systems model reality, or the elements of reality humans perceive. In order to understand and structure their spatial perceptions, humans seem to use more than a single concept of space, and these concepts often are either not formally defined or not able to be implemented. The term **'spatial concept'** is used to describe these notions, which are then formalized and often discretized to form a geometric data model.

A comprehensive description and comparison of geometric data structures is the next major goal. It is hoped that the large number of data models which are heavily driven by implementation, can be reduced to a smaller number of fundamentally necessary traits, for which implementations can be found. This would make comparison of actual systems, communication between geographic information system users and transfer of data between systems much easier as one can then use the reference data model and not be concerned with the conceptually irrelevant differences in the implementations.

Acknowledgements
Comments by Max Egenhofer and anonymous reviewers were helpful and I am grateful for their contribution.

References

Alexandroff, P. (1961). Elementary Concepts of Topology. New York, NY, Dover Publications, Inc.

Beard, K. (1988). Multiple representations from a detailed database: a scheme for automated generalization. Ph.D. Thesis, University of Wisconsin - Madison.

Birkhoff, G. and J. D. Lipson. (1970). "Heterogeneous Algebras." Journal of Combinatorial Theory. **8**: 115-133.

Buchmann, A. , O. Guenther, et al. (1990). Proceedings of the Symposium on the Design and Implementation of Large Spatial databases. Lecture Notes in Computer Science. New York, Springer Verlag.

Chan, K. K. L. and D. White. (1987). Map Algebra: An Object Oriented Implementation. IGIS: The Research Agenda. Arlington, VA,

CODASYL. (1971). Data Base Task Group Report.

Codd, E. F. (1970). "A Relational Model for Large Shared Data Banks." Communications of the ACM. **13**: 377-387.

Codd, E. F. (1982). "Relational Data Base: A Practical Foundation For Productivity." Communications of the ACM. **25** (2): 109-117.

Corbett, J. P. (1979). Topological Principles of Cartography. Bureau of the Census, Department of Commerce, Report 49.

Date, C. J. (1986). An Introduction to Database Systems. Reading, MA, Addison-Wesley Publishing Company.

Diaz, B. M. and S. B. M. Bell. (1986). Spatial Data Processing using Tesseral Methods (collected papers from Tessaral Workshops 1 and 2). Reading, UK, Natural Environment Research Council, Unit for Thematic Information Systems.

Dittrich, K. R. (1988). Advances in Object-Oriented Database Systems---2nd International Workshop on Object-Oriented Database Systems. International Workshop on Object-Oriented Database Systems, September 27-30. Bad Münster am Stein-Ebernburg, FRG, Springer Verlag; Berlin; New York.

Dutton, G. (1979). First International Study Symposium on Topological Data Structures for Geographic Information Systems (1977). Harvard Papers on Geographic Information Systems. Cambridge, MA, Harvard University.

Frank, A. U. (1981). Applications of DBMS to Land Information Systems. Seventh International Conference on Very Large Data Bases VLDB. Cannes, France,

Frank, A. U. (1983a). Datenstrukturen für Landinformationssysteme-semantische, topologische und raeumliche Beziehungen in Daten der Geo-Wissenschaften (Data structures for Land Information Systems-Semantical, Topological and Spatial Relations in Data of Geo-Sciences). Ph.D., Swiss Federal Institute of Technology.

Frank, A. U. (1983b). Storage methods for space related data: The FIELD TREE. Proceedings of Spatial Algorithms for Processing Land Data Symposium. Lincoln Institute for Land Policy.

Frank, A. U. (1984). "Computer assisted cartography: graphics or geometry?" ASCE Journal of Surveying Engineering. **110**(2, Aug.): 159.

Frank, A. U. (1987). Overlay Processing in Spatial Information Systems. AUTO-CARTO 8. Baltimore, MD,

Frank, A. U. and R. Barrera. (1989). The Fieldtree: A Data Structure for Geographic Information Systems. Symposium on the Design and Implementation of Large Spatial Databases. New York, NY, Springer-Verlag.

Frank, A. U. and W. Kuhn. (1986). Cell Graph: A Provable Correct Method for the Storage of Geometry. Second International Symposium on Spatial Data Handling. Seattle, WA, Frank, A. U. and D. M. Mark. (1990). Language Issues for Geographical Information Systems. Geographic Information Systems: Principles and Applications (in press). Longman Co.

Goguen, J. (1989). Modular Algebraic Specification of Some Basic Geometrical Constructions. Geometric Reasoning. Cambridge, MA, MIT Press.

Goodchild, M. F. (1990). A geographical perspective on spatial data models. GIS Design Models and Functionality. Leicester, Midlands Regional Research Laboratory.

Gueting, R. (1988). Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. Advances in Database Technology---EDBT '88, International Conference on Extending Database Technology, Venice, Italy. New York, NY, Springer Verlag.

Guttag, J. V., E. Horowitz and D. R. Musser. (1978). "Abstract Data Types and Software Validation." Comm. ACM. **21**(12): 1048-1064.

Herring, J. R. (1990). TIGRIS: A Data Model for an Object Oriented Geographic Information System. GIS Design Models and Functionality. Leicester, Midlands Regional Research Laboratory.

Herskovits, A. (1987). Spatial Prepositions in English. Cambridge, Massachusetts, Cambridge University Press.

Holroyd, F. (1990). Raster GIS: Models of Raster Encoding. GIS Design Models and Functionality. Leicester, Midlands Regional Research Laboratory.

Johnson, M. (1987). The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. Chicago, University of Chicago Press.

Knuth, D. E. (1973). The Art of Computer Programming. Reading MA, Addison-Wesley.

Lakoff, G. (1987). Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. Chicago, IL, University of Chicago Press.

Lakoff, G. and M. Johnson. (1980). Metaphors We Live By. Chicago, IL, University of Chicago Press.

Mark, D. M. (1989). Cognitive image-schemata for geographic information: Relations to user views and GIS interfaces. GIS/LIS '89. Orlando FL,

Mark, D. M., A. U. Frank, M. J. Egenhofer, S. M. Freundschuh, M. McGranaghan and R. M. White. (1989). Languages of Spatial Relations: Initiative Two Specialist Meeting Report. National Center for Geographic Information and Analysis.

Moellering, H. (1987). A draft proposed standard for digital cartographic data. National Committee for Digital Cartographic Standards.

Neisser, U. (1976). Cognition and reality. San Francisco, W. H. Freeman & Co.

Peuquet, D. J. (1983). "A hybrid structure for the storage and manipulation of very large spatial data sets." Computer Vision, Graphics, and Image Processing. **24**(1): 14-27.

Peuquet, D. J. (1988). Representations of geographic space: toward a conceptual synthesis. Annals of the Association of American Geographers.

Robinson, A. H., R. Sale, J. L. Morrison and P. Muercke. (1984). Elements of cartography. New York, John Wiley & Sons.

Robinson, V. and A. U. Frank. (1984). About different kinds of uncertainty in collections of spatial data. Autocarto 7. Washington, D.C., March 1984,

Samet, H. (1988). "Hierarchical Representations of Collections of Small Rectangles." ACM Computing Surveys. **20**:

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

Samet, H. (1989b). The Design and Analysis of Spatial Data Structures. Reading, MA, Addison-Wesley Publishing Company.

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

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

Tsichiritzis, T. C. and F. H. Lochovsky. (1977). Data Base Management Systems. New York, Academic Press.

Ullman, J. D. (1988). Principles of Database and Knowledgebase Systems Vol 1. Principles of Computer Science Series. Rockville, MD, Computer Science Press.

Zehnder, C. A. (1981). Informationssysteme und Datenbanken. Zurich, Switzerland, Verlag der Fachvereine.