

GODDARD SPACET

11 - 82

79463 - CR

P-31

A Mid-Year Progress Report Submitted to

NASA

Grant NAG 5-798

ADVANCED TECHNIQUES FOR THE
STORAGE AND USE OF VERY LARGE, HETEROGENEOUS
SPATIAL DATABASES

Submitted By

The Pennsylvania State University
114 Kern Building
University Park, PA 16802

for the period
July 1, 1986 - April 1, 1987

Principal Investigator:

Donna J. Peuquet

Associate Professor
Department of Geography

(NASA-CR-181053) ADVANCED TECHNIQUES FOR
THE STORAGE AND USE OF VERY LARGE,
HETEROGENEOUS SPATIAL DATABASES Midyear
Progress Report, 1 Jul. 1986 - 1 Apr. 1987
(Pennsylvania State Univ.) 31 p Avail:

N87-25874

Unclas
G3/82 0079463

A Mid-Year Progress Report Submitted to

NASA

Grant NAG 5-798

ADVANCED TECHNIQUES FOR THE
STORAGE AND USE OF VERY LARGE, HETEROGENEOUS
SPATIAL DATABASES

Submitted By

The Pennsylvania State University
114 Kern Building
University Park, PA 16802

for the period
July 1, 1986 - April 1, 1987

Principal Investigator:

Donna J. Peuquet

Associate Professor
Department of Geography

This document represents a mid-year progress report in the continuing development of a prototype knowledge-based geographic information system in close cooperation with NASA/GSFC personnel. The purpose of this overall project is to investigate and demonstrate the use of advanced methods in order to; 1.) greatly improve the capabilities of GIS technology in handling very large, multi-source collections of spatial data in an efficient manner, and 2.) make these collections of data more accessible and usable for the earth scientist.

Background and Objectives:

A NASA-funded project was begun in 1983 at the University of California at Santa Barbara to investigate the use of new methods to improve the flexibility and overall performance of very large, multi-source, spatial databases. This involved the application of AI concepts with new spatial data representation techniques. This work continues at PSU and currently involves the construction of a prototype knowledge-based geographic information system. This system is being used to empirically test and refine a radically different approach to spatial data representation and processing, as well as a new approach to GIS systems design.

In 1984, NASA/GSFC initiated a complementary research effort in-house, entitled The Intelligent Data Management Project (IDM). The work on this project has to date emphasized intelligent user interface techniques, in contrast to the data storage and management techniques emphasized in the PSU system.

The objective of the research at PSU is thus to continue development of the system, in close cooperation with the efforts at NASA/GSFC, so that compatible approaches are developed that together address a wide range of problems that need to be solved to meet the current and future automated information system needs of the earth scientist.

The approach needed differs from other GIS and image processing systems that have been constructed to date primarily in that they have always employed some form of 'non-intelligent' exhaustive search or explicit look up technique. This necessarily limits:

- 1.) the type of queries that can be answered efficiently to a limited set of anticipated query types that are 'designed-in', and

- 2.) the total volume and range of data types that can be efficiently and economically handled.

The current research represents an attempt to overcome these intrinsic limitations of approaches used in current practice. Preliminary results have revealed that heuristic search and other AI techniques hold much promise as tools for overcoming current efficiency and integration problems being experienced in dealing with the extremely large volumes and variety of spatial data that NASA as a whole must handle.

Given that the spatial data files needed for individual applications or scientific users themselves tend to be large, it also follows that this problem area requires particular attention before NASA can effectively utilize the capabilities of AI technology for higher-level data analysis and decision making.

The specific task associated with this overall research effort is to explore methodologies that will allow the following GIS performance requirements to be satisfied within a single, unified environment;

- 1.) the ability to store and process large, multi-layered, multi-source databases,
- 2.) the ability to query such databases about the existence, locations and properties of complex spatial objects,
- 3.) a level of flexibility that allows the system to be tailored easily to accommodate a wide variety of applications, and
- 4.) the storage of higher-level, derived information while also retaining the original, observational data.

The achievement of these requirements imply the following capabilities within a GIS;

- 1.) the ability to answer a wide range of complex queries posed by the scientist concerning phenomena that may not be explicitly encoded in the database,
- 2.) the use of knowledge-based, non-exhaustive search to limit and control the level of database retrieval needed to answer queries,
- 3.) the use of an extremely efficient and robust database architecture, and
- 4.) the ability to inductively 'learn' new information regarding spatial objects and the relationships between

those objects.

All of these capabilities were incorporated into a proof-of-concept system called KBGIS. This software was completed in early 1986 with funding from the U.S. Geological Survey, NASA and Digital Equipment Corporation. A description of that system is given in the Final Project Report, submitted to NASA for grant NAG 5-369.

Current Status of the Work:

The system currently under construction at PSU, called GeoKnowledge, is based upon the design concepts and overall capabilities demonstrated in KBGIS and represents a continuation of that effort. Work at PSU for the first and current year under the support of NASA Grant NAG 5-798 was originally proposed to consist of the following enhancements to the existing KBGIS system;

- 1.) continuing refinement of the heuristic spatial search structure,
- 2.) investigation of specialized AI tools for use in spatial database applications, and
- 3.) begin development of a graphics interface.

Given that the funding level granted for this work was reduced by more than half of the originally requested level, work on the graphics interface was postponed completely and investigation of specialized AI tools was severely restricted. Work in these two areas were thus included as work elements of a follow-on proposal. The following is a brief summary of the work accomplished to-date.

The priority element, with the concurrence of the NASA technical monitor, was the continuing refinement of the use of higher-level knowledge in efficient, non-exhaustive spatial search of a very-large, heterogeneous database.

Using the KBGIS demonstration system, the strategies used in the search process and the rules used to guide search at all stages of search were empirically examined. It was soon discovered that the slowness and inflexibility experienced in the initial system was due to unexpected interactions of low-level spatial operators. In investigating this problem, it was also quickly realized that a fundamental framework regarding spatial data models did not exist. For both analytical and database applications until now, representational schemes have been developed on an ad-hoc basis using a heuristic approach (often

hardware or language-driven), with little or no consideration for logical consistency or conceptual adequacy.

A formalized framework for the current, or any other system, is seen to provide two benefits:

- 1.) enable the systematic design of flexible and robust data models for large, multi-source data sets with predictable results, and
- 2.) ensure logical and functional consistency of spatial entities and the operators used to manipulate those entities.

The problems encountered were therefore tackled on two levels: the development of a theoretical context, and the development and implementation of new spatial operators couched within that context.

An overall framework was built and used to refine the spatial data model used in the present system and to determine an elemental and consistent set of spatial operators and study their characteristics. Building all higher-level functions from this elemental set with known characteristics avoids the problem of unforeseen interactions. It also allows great flexibility in defining higher-level functions tailored to a wide range of data types and applications. A preliminary description of the characteristics and use of this framework within the present system is attached as Appendix A of this report.

The refined spatial operators have also been developed and implemented within the current system, and a preliminary description of this elemental set of spatial operators is also included as Appendix B of this report.

Both of the above preliminary descriptions are currently being expanded and revised for publication in scholarly journals. These will be included with the final written report for this project. The final report will also provide a unified description of the results of the research.

A demonstration of the complete capabilities of the software developed to-date to be given at NASA/GSFC is being planned at the end of the current year project as an oral report to NASA technical staff. It must be noted that this software will still be in a state of active development as a research tool and is not intended to be a complete or production-level system.

Appendix A

Data Structures for Very Large Geographic Databases
Proceedings, International Workshop on Geographic
Information Systems, Beijing, 1987.

DATA MODELS FOR VERY LARGE GEOGRAPHIC DATABASES

Donna J. Peuquet

Department of Geography
The Pennsylvania State University

ABSTRACT

There is an urgent need to use geographic information systems (GIS) to manage extremely large databases containing data integrated from a number of imagery, cartographic and other sources for an increasing variety of applications. However, current GIS technology has revealed severe shortcomings in meeting these performance requirements.

The cause of this problem is that the spatial data models used in these systems have always been either hardware-driven, such as imagery data, or simple representations of the paper map. In both cases, a number of special characteristics of geographic data have not been taken into account. These characteristics include: First, natural geographic boundaries tend to be very convoluted and irregular. They consequently do not lend themselves to compact representation, and storage of these data can quickly become very large. Second, the data in digital form tend to be incomplete, imprecise and error-prone due to the complexity of the data and the characteristics of the data gathering process. Third, spatial relationships tend to be inexact or application-specific.

The present paper presents a new approach to building geographic data structures that is the basis of a prototype system currently under development. This approach combines Artificial Intelligence techniques with recent developments in spatial data processing techniques to overcome these problems.

INTRODUCTION

The primary bottleneck in the use of geographic information systems in large-scale, real-world applications for many years was that spatial data input was a very slow and expensive process. As a direct result, operational databases tended to be limited in size, regardless of the intended scope of the completed database. Much attention was given to efficient data capture and input, and relatively little to the final form in which the data would be represented.

Due to the advancement of data capture and input techniques and subsequent availability of data from Landsat and other automated data capture devices, this situation has changed dramatically. There is now a rapidly expanding volume and variety of spatial data available in digital form. These data represent a very major investment and an extremely valuable resource which is in demand for an expanding variety of research and decision making applications.

This rapid increase in data availability has caused a major crisis in the handling of these data. Current techniques for conceptually representing and storing spatial data have exhibited severe performance problems. Attempts to integrate the vastly expanded volume and variety of data into new or existing systems have to-date proven extremely difficult, at best.

Much attention has been paid recently in the literature to the development of new methods for representing geographic data in an extremely efficient and flexible manner (Samet, 1984; Van Roessel, 1986). Although each has individual merit, these seem to represent a continuation of the ad-hoc approach toward spatial data modeling that has led to the current situation. It is the author's contention that the only way to overcome the severe efficiency and versatility problems currently being experienced is to develop a unified approach to the design and evaluation of spatial data models that is based on the intrinsic characteristics of geographic data. Such a unified approach should also result in far better predictions of data model performance before implementation.

In the following discussion, the term 'data model' is defined as the conceptual data representation scheme. The term 'data structure', however, refers specifically to the programmable implementation of a data model within the context of lists, pointers, etc.

The purpose of the present paper is to examine how a such a universal approach for spatial data modelling can be developed. In the remainder of the paper, possible insights from existing data modeling techniques developed in a number of disciplines will first be discussed. The basic characteristics of a universal spatial data model will then be given. Finally, the implications and future directions for such a model will be discussed.

TOWARD A NEW APPROACH

The overall geographic information system performance requirements can be summarized as follows: GIS are needed that can;

- 1.) handle extremely large volumes of both coordinate and descriptor data,
- 2.) handle a wide range of data types,
- 3.) accommodate a wide range of queries in varying applications contexts,
- 4.) provide interactive response, and
- 5.) be dynamic, allowing frequent additions and modifications to the database.

This last characteristic means that the database needs to grow and change over time. Included in the third requirement is that the overall flexibility of the data model is capable of accommodating some unforeseen applications.

In light of these requirements, we will now examine a number of data modeling approaches.

'TRADITIONAL' GEOGRAPHIC DATA MODELS

The most universal and well-known representational scheme for geographic phenomena is the paper map. Every cartographic representation implies some conceptual view of the world. Selected geographic phenomena are interpreted by the cartographer in order to visually convey a message. Many styles of cartographic interpretation have evolved, and the cartographer must often take liberties with reality in an ad-hoc manner in order to achieve a desired visual effect.

There have been many digital cartographic data models developed that are direct translations of the analog cartographic document in that they model the map as line-by-line (i.e., vector) representations in digital form. Although they are useful in specific contexts, they have limitations in the faithfulness to which they can represent the original information.

The representation of geographic data captured by remote sensed imagery, on the other hand, has historically been hardware-driven. The form of the data was primarily determined by the characteristics of the sensor, rather than by the characteristics of the phenomena being represented. This was usually raster-scan form. The raster-scan model has the advantage of also being compatible with the hardware/software environments of conventional computers. Many efficient algorithms for processing remote sensed imagery in raster-scan form have been developed as a result. Nevertheless, this data model has proven to be inefficient for the incorporation of cartographic data into an image-derived database. Difficulty in both compatible compaction schemes and higher-level analytical algorithms have been encountered. This may arise from a fundamental difference between the two types of spatial models: Maps are concerned with describing conceptual objects (e.g., lakes, roads), whereas imagery is location-oriented.

DATA MODELS FOR DATABASE MANAGEMENT SYSTEMS

In order to find a better approach for representing geographic information, we can derive insight by studying current techniques initially developed within the field of Database Management Systems (DBMS) for modeling non-spatial data related to business applications (e.g., personnel and inventory). Although the first use of computers for such applications began at approximately the same time as the first use of computers for geographic data, DBMS technology now seems to have progressed to a much more advanced state. Many studies have been done on how to apply the principles of state-of-the-art relational databases to geographic applications [Shapiro & Haralick, 1980; Van Roessel, 1986].

Developments in this field were driven by a need for efficiency in a practical, implementational context. A uniform framework was seen as the means of achieving this. The fundamental rationale in the initial development of the relational database concept was to provide a unified and consistent model for structuring the data with minimal redundancy. The most successful approach developed within the field of DBMS to date is known as the Relational Database Model.

This model is based on the 'relation'. Each relation is simply a table containing a set of individual data entities or observations that are related in some manner. Each row in a relation contains attributes pertaining to an individual element. Each column contains values for a specific attribute for all elements. The relational model is directly derived from the mathematical concept of relations as properties of ordered sequences. For example, the expression $x + y = z$ defines a three-place relation for the set of natural numbers. Much elegance and power of the relational model is derived from one characteristic: Relationships between entities or groups of entities are not explicitly stored, but act as operators on the tables to produce derived relations. These relational operations are specified using either the relational algebra or the relational calculus. This ability to generate derived relations provides users with their own views of the database. The manner in which the relational operators can be used is limited and controlled by a group of built-in rules.

Several inherent shortcomings were soon discovered in this overall model. The two foremost of these were that actual implementations proved too slow for databases of any size and that this model is well-suited only for data with a regular, homogeneous structure. Extensions to the relational model were subsequently developed using techniques developed in the Artificial Intelligence community. These were based on the observation that the relational calculus used in relational database management systems is precisely equivalent to the predicate calculus used for logic programming [Gallaire & Minker, 1978]. The use of a rule-based, graph-theoretic approach has proven to be a powerful mechanism for modeling spatial relationships as operators. Nevertheless, it was seen to be severely limited due to a bewildering number and variation of potential spatial relationships and to a complex of often unpredictable side effects that can be produced by combining these relationships in arbitrary sequences.

The field of Database Management Systems, therefore, has provided a number of valuable concepts for a general model of geographic phenomena, although both geographic theory and direct use of the relational model in its current form are not adequate for this task. The problem of spatial relationships can only be handled by reducing the set of all spatial relationships into a small set of atomic or primitive spatial relationships with known characteristics. From this, formalized rules for combining operations and formulating higher-order relations can be derived systematically.

As a starting point for development of an overall framework for representing geographic phenomena, a robust definition of a data model that has evolved within this field can be employed. This definition can be summarized as follows:

A data model may be defined as a general description of specific sets of entities and the relationships between those sets of entities. An entity is a thing which exists and is distinguishable; i.e., we can tell one entity from another. An entity set is a class of entities that possesses certain common characteristics [Ullman, 1982, pp 12-17].

Given this definition, a chair, a person and a mountain are each individual entities, whereas chairs, people, and mountains are each entity sets. Relationships include such things as 'left of', 'taller 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 mountain may have attributes of size, elevation and geologic strata, among others.

We will now attempt to apply the extended Relational data model approach to the development of a unified conceptual view of geographic space.

A GENERAL FRAMEWORK FOR REPRESENTING GEOGRAPHIC INFORMATION

BASIC COMPONENTS AND CHARACTERISTICS

Key characteristics of geographic phenomenon that need to be taken into consideration in formulating a representational framework for geographic information are:

- 1.) the enumeration of entities, their properties and the relationships between entities tend to be imprecise, incomplete and view dependant,
- 2.) observed or recorded properties of entities can be numerous, and
- 3.) the boundaries of geographic objects tend to be convoluted and irregular.

Adopting the definition of a data model given in the previous section, it is assumed that any geographic data

model can be reduced to the following components;

entities
properties
relationships.

Entities can be grouped into higher-order entities, and both entities and relationships have properties or attributes.

Properties of entities can include general properties such as size, shape, color and height. They may also include domain-specific properties such as geologic strata in the case of mountains. With reference to a specific entity, each known property can be assigned a single value, a range of values, or a group of different values determined on differing measurement scales.

From these characteristics, the method of representation for spatial entities should;

- 1.) allow entities of any level of abstraction to be represented,
- 2.) use generalization, aggregation and successor functions as relational operators between entities and groups of entities, resulting in a conceptually hierarchical structure of entities,
- 3.) allow any number of attributes and more than one value for any attribute for any entity,
- 4.) allow for entities that may overlap
- 5.) allow for measurements at varying degrees of precision.

The hierarchical structure of entities would be defined through the use of abstraction functions as relational operators between entities and groups of entities. These operators would vary to suit the nature of the specific entities involved (i.e., they would need to be knowledge-based and domain-specific). Ultimately, this would constitute a taxonomy of geographic objects with respect to a given context, such as the general example shown in Figure 1.

An important factor to be considered is the manner in which people acquire and use knowledge of the perceived world. All spatial questions can be classified into two basic categories that are logical duals of each other:

- 1.) Given a specific object or objects, what are its

associated properties (one of these properties may be its location or locations)?

2.) What object or objects are present at a given location?

These correspond to object-based and location- or scene-based views, respectively. These primary representation and usage characteristics of geographic information supports the use of a dual structure for modeling spatial phenomena and organizing spatial knowledge, one side being object-based and the other being location-based.

Given a dual structure, it is helpful to slightly refine the definition of the elemental components of a spatial model to the following:

object-based representation

objects
properties
relationships

**location-based
representation**

locations
properties
relation-
ships

In this scheme, locations can also have properties or attributes, such as elevation, temperature, etc. These represent 'primitive' properties, i.e., properties that are directly observable and are not necessarily characteristic of a particular object or objects. Relationships in a location-based context can take on a very special character - these are spatial relationships, such as 'contains', or 'left-of'.

These concepts will now be cast into a more detailed, operationally-oriented structure.

REPRESENTATION OF SPATIAL ENTITIES

There has been much work recently in the field of Artificial Intelligence concerning the representation of knowledge pertaining to individual entities. Central to these representational schemes is the expression of entity definitions in a formal language, such as first-order predicate calculus. This approach allows the use of operators (e.g., and, or, not) in an expression to express a set of constraints that uniquely characterize that object. These are the properties that can be interpreted as the 'valid' or essential properties of that particular object and may include size range, etc.

The set of all objects are implicitly arranged in inter-linked hierarchies, as shown diagrammatically in Figure 1. These hierarchies are defined by the relationships to other objects contained within the object definitions. Such object relationships, for example, include 'is_a' and 'component_of'.

In the location-based representation, locations are discretized into non-overlapping areal cells. Although space is perceived to be continuous, this is a necessary mechanism for recording variations over space in any formalized manner. For the sake of explanation and convenience, we divide our perceived universe in grid fashion into squares of uniform size. We can then logically superimpose increasingly coarser grids in hierarchical fashion to represent the same total area at increasing levels of generalization.

A convenient example of such a structure is the quadtree, as shown in Figure 2. This structure is based upon a recursive subdivision of a square area into four equal subunits. This results in a regular hierarchy of degree four and in cartographic terms produces a variable scale scheme based on powers of 2. This structure may not be the most appropriate for some types of information, but does provide a universally applicable, uniform structure that allows easy association of various types of information for the same areal unit. The quadtree also has been well studied and offers significant implementation advantages, as discussed in Peuquet [1984].

All locational properties can be logically viewed as individual surfaces layered on top of each other. All information pertaining to a single location at any level in the hierarchy (i.e., a node in the quadtree), however, should still be referenced with a single, unique locational index. Such indexing schemes have been discussed for quadtrees in Peuquet, Abel and Smith and others [Peuquet, 1984; Abel & Smith, 1983]. Each location contains information pertaining to each layer (i.e., a single property for that location. For example; property value(s), as well as the name(s) of the specific method(s) used to abstract property values upward through the hierarchy. These methods are known as inheritance rules. This abstraction method may be specific to the particular property and may incorporate higher-level knowledge of the characteristics of that property. Information on how data for that layer are spatially distributed in the descendant, finer-resolution cells representing the same area would also be stored at individual locations throughout the hierarchy.

At the lowest level of the hierarchy, representing the

finest locational resolution are the primitive, observed values. This is not necessarily at the same level in the hierarchy for all properties, in conformance with real-world observation.

RELATIONAL OPERATORS

As previously stated, there are two different types of relational operators in a spatial context;

abstraction relations, and
spatial relations.

Abstraction relations fall into two subtypes, one for combining geographic objects. We can call these taxonomic relations, and include 'is_a' and 'component_of'. These operate on and define the object hierarchy, and they tend to be highly domain-specific. The other subtype combines the values of properties. These include average, mode, maximum, minimum, and any of a multitude of domain-specific aggregation or generalization techniques. Such techniques are well-studied and well-known. They also function on properties pertaining to objects, such as size and shape.

Spatial relations are unique to locational or spatial information. These relations are extremely important but not well-understood in any formal sense. Existing literature in this direction is very sparse and has primarily been done within the field of computer vision [Winston, 1975; Evans, 1968]. In work to date, varying lists of 'basic' spatial relations have been given. Algorithmic models for these relations have been very simple and limited to the domain of regular geometric figures.

Since this seems to be a major missing element that is essential to the definition of any formalized representation of geographic knowledge, we will now try to provide some insights into this area in a geographic context.

On the basis of work performed by the author [Peuquet, 1984; Smith & Peuquet, 1985], it seems that all spatial relationships can be stated in terms of the following primitives;

boolean set operations
distance
direction

For example, the higher-order spatial relation 'nearest

neighbor' can be expressed as a series of relative distance relationships. Similarly, 'between' can be expressed as a specific and limited combination of possible direction relationships. 'Touching' or 'adjacent' can be expressed as a special case of distance, where the distance between one object and a second object equals zero at one or more locations and is never less than zero. 'Left-of', 'right of', 'above' and 'below' are specific instances of the same relational concept (i.e., direction) in that the same model holds for all. A model for 'left of' becomes a model for 'right of' after performing a 180 degree coordinate rotation on the data.

This means that developing an understanding of spatial relations in a formal, theoretical context is a much more tenable task than had been previously assumed, as only three spatial relationships, their characteristics and interactions need to be formally defined. All other spatial relations can then be defined in terms of these primitive relations and a set of combinatorial integrity rules. This is also particularly encouraging in the derivation of a complete and robust framework.

Recent research in deriving robust models for each of the spatial relational operators above shows that the further development of such operators holds promise [Peuquet & Ci-Xiang, 1987; Peuquet, 1987];

- 1.) by virtue of the small number of primitive relational operators, and
- 2.) because some understanding and adequate algorithmic approaches for primitives already exists.

It is easily seen that there is a wide variation in how certain aspects of these primitives may be defined. Further verification that the three operators listed above do in fact comprise the set of primitive spatial relational operators needs to be undertaken.

SUMMARY AND FUTURE DIRECTIONS

The elements and characteristics of a formalized conceptual framework has been discussed and an example of a structure for representing spatial knowledge has been described. From this it seems that the overall characteristics suggested (e.g., hierarchical structure, separation of location-based and object-based views and the ability to store knowledge at variable levels of completeness and precision), draws great support on the basis of an

agreement of findings among related disciplines. Given a significant amount of research in the recent past, powerful methods for appropriately representing both locational and object views conforming to these characteristics are shown to be available.

This discussion, however, hints at many other issues. Several issues, unique to the geographic context, remain as major obstacles in using this as a functional knowledge representation for practical applications and prime areas for further theoretical research. The first is in refining the definitions and understanding of primitive spatial relationships and how they interact so that, at minimum, a relational inference structure can be developed. This is needed before these primitives can be used in formalizing definitions of higher-order relations.

An obvious issue that has not been explicitly stated so far in the present discussion: There are wide variations in semantic meanings of spatial relations in natural language expressions. The first task is certainly to derive canonical geometric description functions for primitives and a mechanism for combining them in a strict, formalized manner. With this in hand, the problem of defining semantic deviations in context from these 'ideal' forms, including definition of approximations, could be more easily handled.

REFERENCES

- Abel, D. & J. Smith, 1983. "A Data Structure and Algorithm Based on a Linear Key for a Rectangle Retrieval Problem." Computer Vision, Graphics, and Image Processing. Vol. 24, pp. 4-14.
- Evans, T., 1968. "A Heuristic Program to Solve Geometric Analogy Problems." in Semantic Information Processing, M. Minsky (ed.) Cambridge: The MIT Press.
- Gallaire, H. & J. Minker, 1978. Logic and Databases. New York: Plenum Press.
- Peuquet, D., 1984, "Data Structures for a Knowledge-Based Geographic Information System." Proceedings of the International Symposium on Spatial Data Handling. Zurich, pp. 372-391.
- Peuquet, D., 1987. "An Efficient Distance Operator for-Quadrees." unpublished report, Dept of Geography, The Pennsylvania State University.
- Peuquet, D. & Z. Ci-Xiang, 1987. "An Algorithm to Determine the Directional Relationship between Arbitrarily-Shaped Polygons in the Plane." Pattern Recognition, forthcoming.
- Samet, H., 1984. "The Quadtree and Related Hierarchical Data Structures." ACM Computing Surveys, Vol. 16, pp. 187-260.
- Shapiro, L. and R. Haralick, 1980. "A Spatial Data Structure." Geoprocessing. Vol. 1, pp. 313-337.
- Smith, T. & D. Peuquet, 1985. "Control of Spatial Search for Complex Queries in a Knowledge-Based Geographic Information System" Proceedings, International Conference on Advanced Techniques for Monitoring and Processing Global Environmental Data.
- Ullman, J., 1982. Principles of Database Systems. Rockville, Md.: Computer Science Press.
- Van Roessel, J., 1986. "Design of a Spatial Data Structure Using the Relational Normal Forms." Proceedings, Second International Symposium on Spatial Data Handling. Zurich, pp. 251-272.
- Winston, P., 1975. "Learning Structural Descriptions from

Examples." in The Psychology of Computer Vision, P.
Winston (ed.) New York: McGraw Hill.

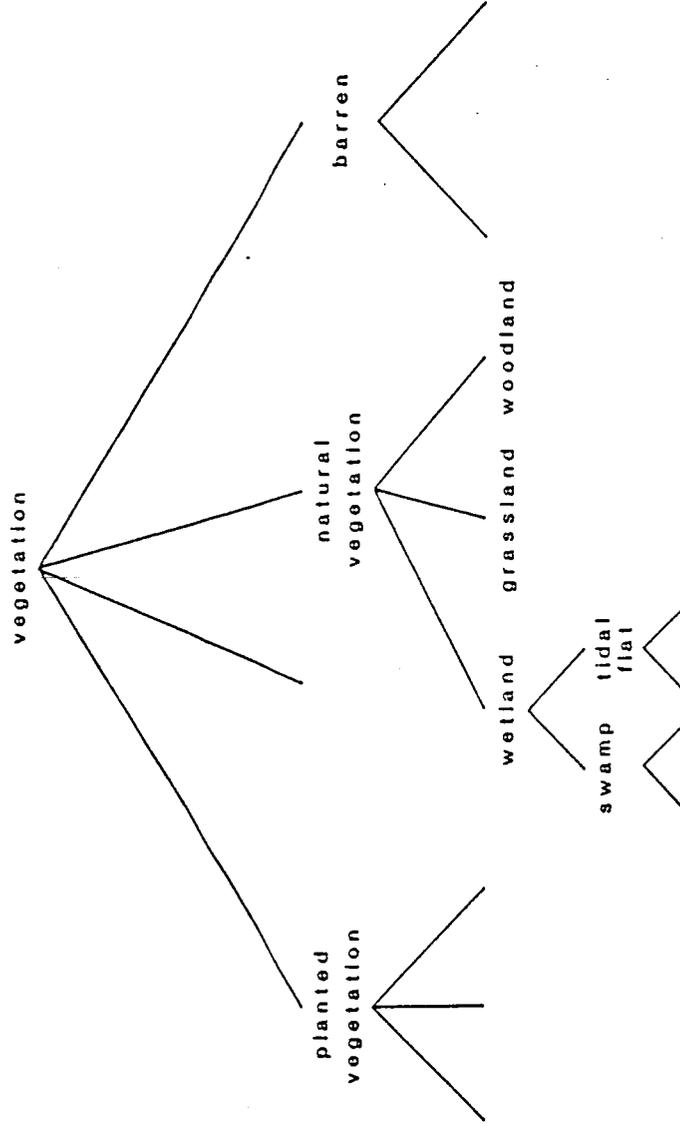
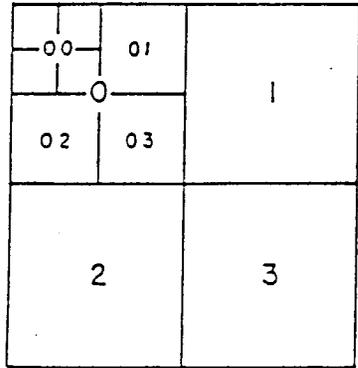


Figure 1: A simple object tree



- ONE QUADTREE FOR EACH VARIABLE
- DIRECT ADDRESSING

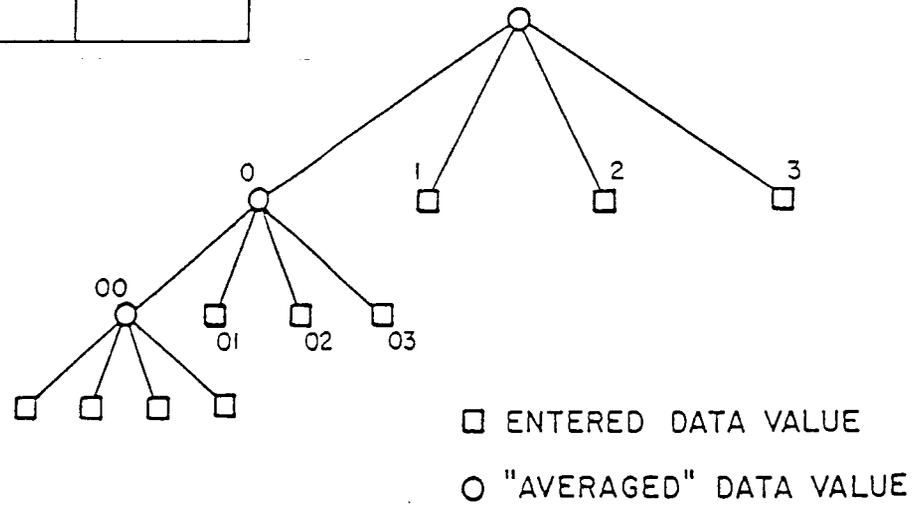


Figure 2: General quadtree structure: numbers show a hierarchical locational indexing scheme

Appendix B

The Use of Spatial Relationships to Aid Spatial Database Retrieval, Proceesings, Second International Symposium on Spatial Data Handling, Seattle, 1986.

THE USE OF SPATIAL RELATIONSHIPS TO AID SPATIAL DATABASE RETRIEVAL

Donna J. Peuquet
Department of Geography
The Pennsylvania State University
University Park, PA 16802
U.S.A.

Introduction

The primary bottleneck in the use of geographic information systems in large-scale, real-world applications for many years was that spatial data input was a very slow and expensive process. As a direct result, operational databases tended to be limited in size, regardless of the intended scope of the completed database. With gradual accumulation over time and with the advent of direct digital data capture and mass digitizing systems, volumes of data encountered in geographic databases can now be extremely large. Experience with GIS technology has also increased user sophistication and expectations with regard to the range of applications, complexity of queries and response time.

Because of these factors, better methods for storing and retrieving spatial data need to be used in order to avoid severe performance problems. Much attention has been paid recently in the literature to the development of new methods for representing geographic data in an extremely efficient and flexible manner (Samet, 1984; Peuquet, 1984). Nevertheless, little is being done on the necessary counterpart of this work; to increase the efficiency and flexibility of search techniques that operate on these geographical data models. This is therefore the topic of the current paper.

The remainder of the paper will be organized as follows; The nature of spatial queries and their components will be discussed, first from the point of view of a user's logical interpretation and the query language, and then from an implementation perspective for answering such queries. The implications that spatial search has on the selection of a specific database model will next be examined. Based on this, necessary characteristics of database operators for efficient search are derived. A model for each primitive relationship will then be presented in general algorithmic terms. Finally, the implications given the requirements for spatial relationship operators and the current understanding of these operators will be discussed.

The Nature of Spatial Queries

Spatial queries, as a user might pose them, can become very complex. An example of such a query would be:

Find the locations of all undeveloped parcels of land greater than 10 acres in size, west of Washington, D.C. but not more than 15 miles away.

For the purpose of database retrieval, any spatially-oriented database query can be generally described as:

find location(s) of the given spatial object which satisfy the given constraints.

Thus, in the example above,

undeveloped
greater than 10 acres
west of Washington, D.C.
not more than 15 miles away (from Washington D.C.)

are the specified constraints that limit the number of occurrences of 'parcels of land' that can logically satisfy the above query.

The possible spatial constraints in such a query can be any combination of two basic types;

- 1.) spatial properties that can be used to describe individual objects, or
- 2.) spatial relationships between objects.

Spatial characteristics of individual objects include size (area, length), centroid, shape, convolutedness and texture. Examples of spatial relationships between objects include distance, nearest neighbor and containment.

If users can input complex queries directly into a GIS, it follows that these logical relationships between objects should also be used as search constraints to limit the areas of a spatial database that need to be examined. This would result in a substantial improvement in search efficiency. Spatial relationships between objects, however, are not well understood in some cases, and efficient algorithms for them have not been developed to a significant degree. An additional problem that follows from this is that the complexity of spatial queries that can be directly input to current operational systems tend to be severely limited. Only a single spatial relationship is usually allowable in a single query, although multiple data layers may be involved. The use of one or more boolean set operations is the one possible

exception.

The spatial properties of individual spatial objects, such as size, are very well understood and can be precisely defined in geometric terms. It is nevertheless difficult to use characteristics of individual objects to aid spatial search unless a-priori knowledge concerning their distribution has been stored in some manner.

The Direct Use of Spatial Relationships in Spatial Database Search -

At the current time, the only constraint commonly used for spatial search for which efficient algorithmic approaches exist is range searching (i.e., direct spatial retrieval of a rectangular area within a given maximum and minimum x-y range) (Preparata and Shamos, 1985; Bentley and Friedman, 1978).

Researchers who have previously examined spatial relationships have given varying lists of basic relationships and have offered some simple models. Freeman, for example, lists 'between', 'touching', 'left of', 'right of', 'above' and 'below', among a total of thirteen. On the basis of work performed by the author within a larger research context (Peuquet, 1984; Smith and Pazner, 1984), it seems that all spatial relationships can be stated in terms of the following primitives;

- boolean set operations (and, or, not)
- distance
- direction

For example, the higher-order spatial relation 'nearest neighbor' can be expressed as a series of relative distance relationships. Similarly, 'between' can be expressed as a specific and limited combination of possible direction relationships. 'Touching' would be a special case of distance, where the distance between one object and a second object equals zero at a single location and is never less than zero (i.e., crossing over to the inside). Similarly, 'adjacent' would apply if the distance between the two objects for some number of consecutive locations equals zero but is never less than zero. 'Left of', 'right of', 'above' and 'below' are specific instances of the same relational concept (i.e., 'direction') in that the same model holds for all. A model for 'left of' becomes a model for 'right of' after performing a 180 degree coordinate rotation on the data.

Although it may not be ultimately desirable to implement all spatial relationships within a GIS as literal combinations of only these primitive relationships, it provides a simple basis on which to systematically build a more thorough understanding of all spatial relationships and of how they interact.

Given this very limited number of spatial relationship

primitives to build upon, it also becomes feasible to construct a spatial language in which any spatial query of arbitrary complexity can be expressed. It would seem reasonable that such a language would be an extension of the relational calculus or first order predicate calculus. In this context, spatial properties can be expressed as any other non-spatial property, and spatial relationships could be expressed as additional operators between objects. The syntax of such spatial languages have already been explored by a number of researchers, such as Shapiro and Engineer (1984) and Smith and Pazner (1984).

The limiting of search areas by the use of spatial relationships (i.e., spatial relationship constraints) can be affected in two ways:

- 1.) The incorporation of spatial relationship constraints into the search process itself would have the direct effect of avoiding areas that have little or no chance of satisfying a given spatial relationship.
- 2.) An understanding of how spatial relationships interact can result in powerful query optimization techniques aimed toward; (a.) avoiding areas at all stages of the query that have little or no chance of satisfying the combined constraints, and (b.) ordering the satisfaction of individual search constraints in such a way as to eliminate a maximal amount of area to be searched as early in the process as possible.

The second factor is obviously dependent on the first, so it is the development of individual spatial relational operators that is the focus of the current discussion. Before the spatial relationships themselves can be discussed, some characteristics of the possible data model alternatives with relation to spatial search will now be briefly examined.

The Relation between Spatial Retrieval and Data Models

For an algorithmic approach to implement a given spatial constraint for search in a large, heterogeneous database, it is assumed that such an approach must not only be non-exhaustive, it must avoid unlikely areas to a maximal degree. One algorithmic strategy that seems to have significant power in this regard in a spatial data context is divide and conquer (Preparata and Shamos, 1985). Second, the data model on which the algorithm operates must be flexible enough to accommodate a wide variety of functions and applications. Thus, specialized data models that facilitate only a specific and narrow range of tasks are not appropriate. A prime example of this is the use of K-d trees for facilitating nearest-neighbor and adjacency operations (Samet, 1984).

Either of the two basic types of spatial data models, vector or tessellated, can be used to answer queries involving spatial constraints. Nevertheless, since vector data models are logically organized by object (e.g., lakes, roads or cities) and tessellated models are organized spatially (i.e., by geographic location), there are implications as to their respective potentials for performance in spatially-oriented queries.

Vector data models can be used efficiently for a limited range of anticipated queries involving spatial relationships. This usually entails either explicitly recording spatial relationships, such as adjacency, directly into the database as data elements or sequencing the entity records according to some spatial ordering scheme. Selected spatial queries can thus be designed into the database. If there are multiple spatial interrelationships to be taken into account, this approach can quickly become cumbersome. Unanticipated (i.e., not built-in) spatial relationships prove to be extremely inefficient, since they would likely entail an exhaustive test of all coordinates in the database, or a preprocessing step to impose a specific spatial ordering on-the-fly.

One attempt to overcome this problem is the use of a relational database approach for object-oriented data models. The data model in relational database systems consists of a set of relationships (of various sorts) explicitly represented as tables and a set of data integrity rules. By use of these rules, new spatial or non-spatial relationships can be defined and stored within the database to accommodate new types of queries. However, there are no controls for judging when a new relation should not be explicitly stored or when a previously stored relation should be deleted from the database. There is still the inherent limitation of this approach to handle fuzzy relationships and to handle a wide range of unanticipated queries, particularly for large volumes of data. Another problem with this approach within the context of large databases is that searches are exhaustive. These inherent limitations of the relational database concept has been discussed by Codd (1982).

Tessellated data models generally allow for much more efficient handling of complex and unanticipated spatial queries. The reason for this is that, by their intrinsic spatial ordering, all potential spatial relationships are implicitly contained in the model. Any specific spatial relationship or combination of relationships between objects that are represented in a tessellated data model can be directly computed. In this sense, spatial relationships act as operators on a tessellated structure.

Regular hierarchical tessellated data models have the added advantage of being particularly suited to non-exhaustive, divide-and-conquer algorithmic approaches. Search algorithms can very quickly narrow-in on the desired spatial areas using the hierarchical, recursive subdivision of space.

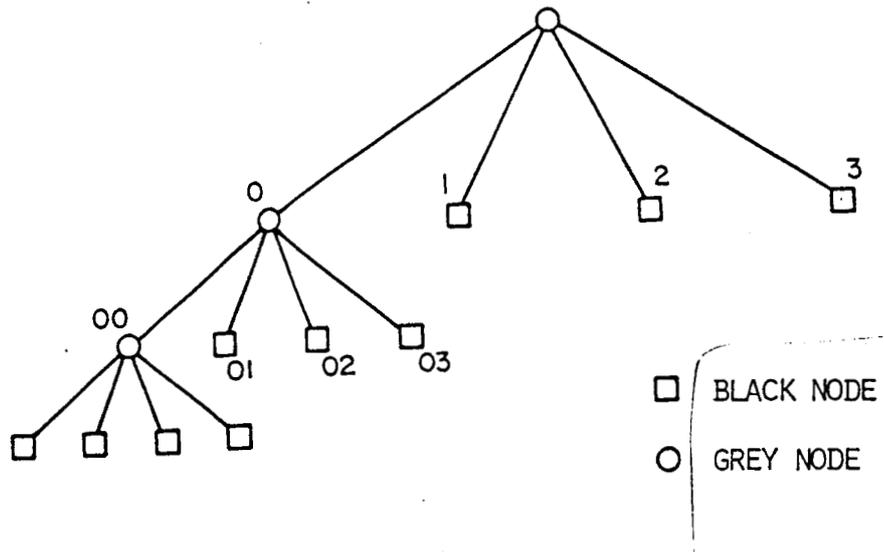
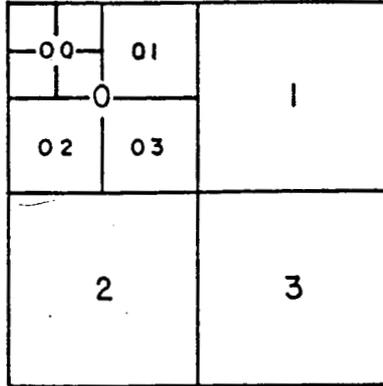


Figure 1. General Quadtree Structure

Intersection -

The quadtree set intersection algorithm of Schneier involves traversing two trees in parallel and selecting the appropriate action for one of only three conditions wherever the traversal reaches a leaf: If a black node is encountered in one tree and the corresponding node in the other tree is also black, then the corresponding node in the resultant tree is also black. If one is black and one is white, then the node in the resultant tree will be white. If a black node is encountered in one tree and the corresponding node in the other tree is grey, the corresponding node in the resultant tree is grey and the structure (i.e., the distribution of black, white and grey nodes with regard to the data value of interest) of the entire subtree below that node is also copied to the resultant tree. Finally, if both nodes encountered in the two input trees are grey, then the entire process is repeated recursively for the descendant nodes.

Union -

The quadtree union algorithm is very similar to the Intersection algorithm. Again, both input trees are traversed in parallel. If a black node is encountered in either of the input trees, the corresponding node in the resultant tree is black. If one node is white and the corresponding node in the other tree is grey, the corresponding node in the resultant tree is grey and the structure of the entire subtree below that node in the input tree with the grey node is also copied to the resultant tree. Finally, if both nodes encountered in the two input trees are grey, then the entire process is repeated recursively for the descendant nodes.

Complement -

The logical complement is a very simple process involving a single input quadtree. The operation involves changing all black nodes to white, and white nodes to black. Grey nodes and the overall structure of the tree do not change.

All of these operators can be combined for multiple overlays following the usual algebraic rules.

distance -

A distance operator that calculates minimum distance between two spatial objects can be developed that takes advantage of the hierarchical quadtree structure. This can be done for polygons with three basic steps:

- 1.) Find the smallest common quadrant that completely encloses both polygons.
- 2.) recursively subdivide the quadrant until the polygons (or portions of two polygons) occur in separate

- quadrants. These quadrants will always be adjacent to each other.
- 3.) calculate minimum distance between the two polygon bounds for each pair of quads and take the minimum distance among all pairs.

Direction -

A special problem is present in the case of relative direction that is probably the reason that the current use of direction as a spatial query constraint is limited to a crude approximation. This problem is that relative size, distance and the shapes of the two objects influence the directional relationship as perceived by humans. The rigidity of the interpretation can also be influenced by the application. A model for relative direction is therefore very difficult to encode.

Unlike other spatial relationships such as distance or adjacency, the directional relationship between two polygons (e.g., left, above, beside, east, north) is a fuzzy concept and is thus often dependent on human interpretation. The problem is also made more complex in the case of arbitrary polygons because of the effects that relative size, distance and shape have on the perceived directional relationship. A model for direction is consequently difficult to derive except in a very generalized form and for simple geometric objects.

A number of researchers have offered insights into the semantics of this and other spatial relations, most notably Freeman (1973), Winston (1975), Evans (1968) and Haar (1976). Their models of the directional relationship, limited primarily to points and squares, have recently been integrated and extended to arbitrarily-shaped objects (Peuquet, 1986).

Some of the basic characteristics of the relationship, however, seem to cause significant obstacles in the development of a hierarchical divide-and-conquer approach except for more than a few limited cases. For example, a simple and straightforward algorithm for quadtrees to determine relative position in eight discretized directions is obvious if the two polygons are located in adjoining quadrants. As can be seen in Figure 2, a directional determination can be made regardless of the relative size of the two polygons. The use of the quadtree structure, however, breaks down when the two polygons are at some distance from each other so that they are not in adjacent quad blocks. The problem comes from a basic characteristic of the directional relation: The area of acceptance for any given direction increases with distance. This implies that any directional model that holds its discriminatory power for arbitrary distances in some way must incorporate a triangular geometry, as shown in Figure 3. Unfortunately, this is not compatible with the quadtree cell structure.

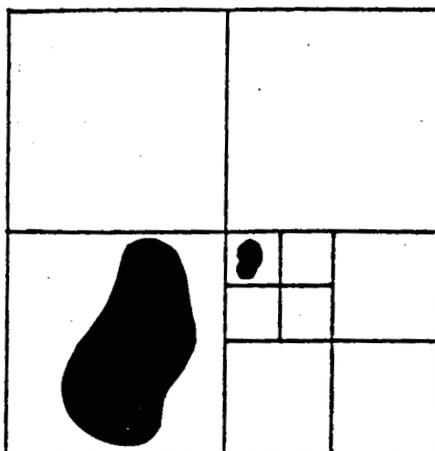


Figure 2. The smaller polygon can be said to be east of the larger polygon as long as it is completely contained within the quadrant adjacent to and of equal size to the quadrant containing the larger polygon.

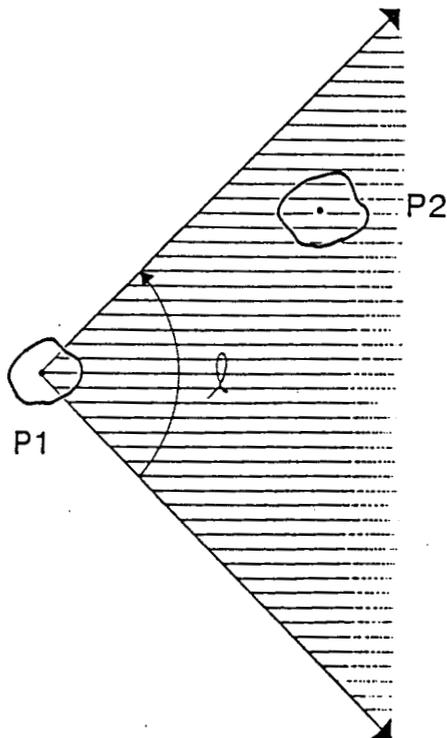


Figure 3. A triangular 'area of acceptance' for a given relative direction takes the effect of distance into account.

Summary and Future Directions -

Using spatial relationships to aid in efficient spatial database retrieval would substantially increase the flexibility, efficiency and overall capacity of GIS. In order to achieve this, we must first have an understanding of these spatial relationships and how they interact with one another. Toward this objective, the primitive spatial relationships must first be identified. These are the spatial relationships that cannot be defined in terms of other spatial relationships. These relationships, in turn, can be used to define all other possible spatial relationships. Based on research performed in a larger context, it was asserted that there are only three such primitive spatial relationships.

On the surface, this holds promise as an easy task by virtue of this small number. We have shown that there are algorithmic approaches for all of these primitives. Upon further investigation, it is soon seen, however, that there is a wide variation in our understanding of these primitives. There are also a number of problems that need to be solved that go beyond shortcomings of specific algorithms. The following is one of the more difficult such problems.

Distance and direction are normally defined as binary operators, as opposed to set operators. Models developed for these relationships, and subsequently algorithms derived from these models, thus assume the presence of only two objects. This is often not the situation in spatial database queries. For example, a typical query may be; "Find all nuclear power plants within 50 miles of any urbanized area within the U.S." Here, what is implied is a set operator that compares the set of all nuclear power plants with the set of all urbanized areas. An area for further research is therefore how to extend our current binary models of primitive spatial relationships so that they can be efficiently applied to multiple occurrences of the two types of objects.

References

- Bentley, J. and J. Friedman, 1978. "Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces." IEEE Transactions on Computers. Vol. C-27, pp. 97-105.
- Codd, E.F., 1982. "Relational Database: A Practical Foundation for Productivity." Communications of the ACM. Vol. 25, pp. 109-117.
- Evans, T.G., 1968. "A Heuristic Program to Solve Geometric Analogy Problems," in Semantic Information Processing. M. Minsky, ed. MIT Press, Cambridge, MA.
- 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.
- Freeman, J., 1973. "The Modeling of Spatial Relations." Computer Graphics and Image Processing. Vol. 4, pp. 156-171.
- Haar, R., 1976. Computational Models of Spatial Relations. Technical Report TR-478 MCS-72-0361-10, Department of Computer Science, University of Maryland.
- Hunter, G. and K. Steiglitz, 1979. "Operations on Images Using Quadtrees." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 1, pp. 145-153.
- Klinger, A., 1971. "Patterns and Search Statistics." in Optimizing Methods in Statistics. J.S. Rustagi, ed., pp. 303-337. New York, Academic Press.
- Klinger, A. and C. Dyer, 1976. "Experiments on Picture Representation Using Regular Decomposition." Computer Graphics and Image Processing. Vol. 5, pp. 68-105.
- Peuquet, D.J., 1984. "Data Structures for a Knowledge-Based Geographic Information System." Proceedings of the International Symposium on Spatial Data Handling, Zurich, pp. 372-391.
- Peuquet, D.J., forthcoming. "An Algorithm to Determine the Directional Relationship between Arbitrarily-Shaped Polygons in the Plane." Pattern Recognition.
- Preparata, F.P. and M.I. Shamos, 1985. Computational Geometry - An Introduction, Springer-Verlag, New York.
- Samet, H., 1984. "The 'Quadtree and Related Hierarchical Data Structures.'" ACM Computing Surveys. Vol. 16, pp. 187-260.

- Shapiro, L. and S. Engineer, 1984. "A Query Language for a Spatial Information System." Proceedings, XV ISPRS Congress, Rio De Janeiro, pp. 450-459.
- Shneider, M., 1981. "Calculations of Geometric Properties Using Quadtrees." Computer Graphics and Image Processing, Vol. 17, pp. 296-304.
- Smith, T.R. and M. Pazner, 1984. "Knowledge-Based Control of Search and Learning in a Large-Scale GIS." Proceedings of the International Symposium on Spatial Data Handling, Zurich, pp. 498-519.
- Winston, P.H., 1975. "Learning Structural Descriptions from Examples." The Psychology of Computer Vision, P. Winston, ed., Mc Graw Hill Book Company, New York, N.Y.