Databases for GIS

Claudia Bauzer Medeiros
cmbm@dcc.unicamp.br
DCC - IMECC - UNICAMP
13081-970 Campinas SP Brazil

Fatima Pires
fpires@dcc.unicamp.br
DCC - IMECC - UNICAMP
13081-970 Campinas SP Brazil

1 Introduction

The problems of discussing *Geographic Information Systems* – GIS, for short – begin with defining this term. There are countless definitions for GIS, each based on the type of user and application domain [43]. The more general definition [24] would be "a digital information system whose records are somehow geographically referenced". For more precise definitions, one may emphasize their functional capabilities (e.g., that GIS capture and process spatial data) or the applications supported (linking the definition to the type of problem solved). Others stress the fact that GIS are ultimately tools to be used for decision support. Last but not least, a GIS is database-dependent ("a database system that supports management of spatial data").

Depending on the definition, different issues are considered. This paper analyses database support for GIS and takes into account the following properties: GIS perform data management and retrieval operations for *georeferenced data*; such data is time and space specific; the data that must be integrated into GIS comes in distinct formats, from different sources and geographic locations, and is captured by various types of devices; it occupies considerable amounts of space and requires specialized analysis and output formatting operations, not available in commercial database systems.

The term *georeferenced* refers to data about geographic phenomena associated with its location, spatially referenced to the Earth [11]. Georeferenced entities are characterized by their specific properties (such as density) and their spatial relationships with other entities (e.g., distance). Given the nature of georeferenced data, present DBMS facilities need to be extended to provide adequate support to GIS applications.

Literature on GIS is mostly of two kinds: research performed by computer scientists, and research conducted by end-users (e.g., cartographers, geographers or social scientists). The sections that follow discuss some of the issues considered by these researchers.

In the database community, GIS are primarily associated with *spatial databases*, and therefore a large amount of the research effort in databases for GIS is related to spatial structures and access methods (see section 7). Research has also considered data modelling [69, 25, 56], query language constructs and optimization (see sections 5 and 6), temporal data support [64, 38, 54], and experimenting with existing DBMS to manage georeferenced data (section 8). Database specialists often ignore many concerns that underly end-users' applications, such as data accuracy evaluation, feature generalization, fuzziness of geometric boundaries or need for versatile data analysis tools.

End-users are concerned with GIS functional issues There has been intensive work on data collection procedures and error elimination (see section 2); representation of geographic reality (section 3); operators and functions for data analysis and result interpretation (section 4); and output formatting (section 5). Users assume a database to be a fundamental part of GIS, but the definition of what is a "database system" is very fuzzy. The term is applied to a variety of notions, from a set of unrelated flat files which are accessed by a georeferenced data handler to a real DBMS which supports an extended query language and has a (cartographic) interactive interface.

2 GIS applications and data

GIS applications involve several domains of knowledge. Examples are [5]: urban planning, route optimization, public utility network management, demography, cartography, agriculture, natural resources administration, coastal monitoring, fire and epidemics control. Each type of application deals with different features, scales and spatio-temporal properties.

[43] distinguishes three main categories of GIS applications: *socioeconomic* applications (covering assembly and analysis of spatial data about the land, the people

and infrastructure); *environmental* applications; and *management* issues for the other two types of applications. Another canonical division (usually assumed by GIS vendors) is based on the geographic scale with which the application is concerned. AM/FM(automated mapping/facilities management, where cadastral applications play a major role) considers data on larger scales that range roughly from 1:500 to 1:20.000. Environmental applications are those where scales are much smaller (e.g., down to 1:5.000.000). These taxonomies are not absolute, since environmental planners may also use cadastral data, and facility management may require environmental studies.

Geographic data has four main characteristics [5]: its geographic position (coordinates); its attributes (data values); its topological relationships; and its time components. Once stored into a GIS, this data can be classified into three main categories [48]:

- *conventional data* – traditional alphanumeric attributes, handled by conventional DBMS;
- *spatial data* – attributes that describe the geometry and location of geographic phenomena. Spatial data has geometrical (e.g., size) and topological (e.g., connectivity) properties.
- *pictorial data* – attributes that store images (e.g., photos) and are managed by image processing functions.

The questions to be asked from a GIS are:

- *What* kind of data is this?
- *Where* does data refer to?
- *When* does data refer to?
- How *accurate* is it?
- *Who* is posing the questions?

The first three questions reflect the spatio-temporal nature of georeferenced data. The fourth question permeates all work on data capture, editing and transfer, and reflects the fact that storing data about geographic phenomena for computer processing usually requires discretization of nature, which introduces errors. Finally, the last question refers to the fact that GIS are ultimately planning tools for decision support. Thus, different users may pose the same question but will want to be shown answers tailored to their level of need.

3 Data Models

Traditional data modelling techniques are not adequate for dealing with geographic information. Difficulties arise from the fact that most geographic data must be considered with respect to the location where it is valid, the time of the observation and its accuracy.

The modelling of georeferenced data by end-users is associated with different perceptions of the world: the *field* model and the *object* model [23, 24]. These models are mapped into different structures: *tesseral* and *vector*. Discussions on the appropriateness of using one or the other generated the so-called "raster-vector debate" [15].

The *field* view sees the world as a continuous surface (layer) over which features vary in a continuous distribution (e.g., atmospheric pressure). Each layer corresponds to a different *theme* (vegetation, soil). Entities are created in the modelling process and do not exist independently. Rather, they are used to partition the field in areas (e.g., by soil type). Emphasis is on contents of these areas, rather than their boundaries.

The *object* view treats the world as a surface littered with recognizable objects, which exist independent of any definition (e.g., a given river). In this model, two objects can occupy the same place (e.g., a beach on the river bank). Database entities correspond to these recognizable objects.

Field data is processed in tesseral format (spatial objects described as polygonal units of space – cells – in a matrix). Each cell contains one thematic value (i.e., there cannot be two types of soil for a given cell). Cells may have different shapes; square cells are called *pixels*. The *raster* format (which is often used as the generic name for tesseral data) is just one special type of tesselation with rectangular grid format, organized in line scan order. In this case, coordinates are not stored, but rather derived by the position of the cells in the scan order.

Object data is processed as points, lines and polygons (the *vector* format model), using lists of coordinate pairs. Boundaries of regions are stored precisely, and several attributes can be associated to a single element. *Networks* are a special case of vector data, where elements are sets of links and nodes. They are used for facility management and network analysis (e.g., in transportation or hidrology). This type of format is usually more adequate for representing man-made artifacts (e.g., bridges) and in AM/FM, whereas the field model is adopted mostly in environmental applications.

4 Spatial Operators and Relationships

Little has been done to identify and formalize GIS key functional requirements, in terms of basic spatial operators and supported relationships. A considerable effort has been undertaken to process and store georeferenced data. However, available systems lack sophisticated analysis facilities [21]. The problem is largely due to the wide spectrum of application domains and variety of users, who have different types of expectations and views of the basic facilities GIS should provide [43]: the ***cartography*** view expects services in terms of map processing and display systems; the ***database*** view stresses the need for database support, but does not count on sophisticated data analysis functions; the latter are the focus of the ***spatial analysis*** view.

Spatial relationships (e.g., distance, adjacency, containment) and operators depend on factors such as scale, time, point of view and preciseness of their specification (e.g., [51, 34]). [57] contains one of the first comprehensive attempts to try to formalize GIS functional requirements, expressed in a temporal logic language. As pointed out in this study, in the absence of unified theories of time, space and accuracy, distinct GIS adopt different procedures to reason about these factors. Thus, it is impossible to establish a fixed, standard, set of rules or procedures to deal with them.

The absence of a basic set of primitive spatial operators and relationships is reflected by the diversity of GIS query language constructs. The definition of such a set is one of the open issues for the end-users' community: each type of application requires different functions. However, all such functions can be broadly classified into [42]:

a) capture, transfer, validation and editing – this is the most expensive part of GIS creation. The system must take into account errors that occur in data capture and transformation, and provide users with means of determining the accuracy of answers.

b) store and structure – these determine (and are dependent on) the type of operations that are to be later allowed. The raster-vector debate is intimately connected with the choice of these functions [15].

c) restructure, generalize and transform – these allow the modification of stored data, either by conversion of data formats, or by scale changing, rotation, translation and aggregation.

d) query and analysis – spatial queries ultimately involve some data analysis function. Problems range from selection of georeferenced entities to their display.

Even when relationships and operators are precisely defined, their implementation is problematic. Most such functions are computed at query processing time, and involve going through large data sets, where data has different scales and sometimes fuzzy boundaries. In some cases, the use of spatial indices is possible (see section 7). In other cases, it is claimed that object-oriented systems can help, since composition may allow implicit indication of some relationships (e.g., containment).

5 Interfaces

GIS interfaces usually support two types of interaction mechanisms, discussed in this section:

- textual query languages
- interactive manipulation of geographic elements

Two issues must be considered: the adequacy of these mechanisms for spatio-temporal querying, and the actual output provided.

The initial cost of input processing is commonly 5 to 10 times the cost of GIS hardware and software [5]. During data input, the spatial and non-spatial components of georeferenced entities must be entered and correctly linked. Many of the most difficult issues confronting GIS applications, and especially those based on map data, arise from data preparation steps – e.g., problems associated with map interpretation, coordinate system incompatibilities or image analysis [43].

Output processing involves sophisticated display procedures and requires knowledge of computer graphics and image processing. One important output requirement that has not yet been satisfactorily handled is to allow tailoring visualizations of the same data so that users can switch between several data displays of the same georeferenced entities.

5.1 Textual queries: extended SQL

Different extensions have been proposed to SQL to transform it into a spatial language. The unsuitability of this approach is discussed in [18]. The following shortcomings are pointed out:

- difficulties of incorporating spatial concepts of graphical specification and display into an SQL (flat table) framework;

- the lack of power of the relational model to support qualitative answers, knowledge and meta-data queries.

[18] points out that any spatial SQL extension is a short term solution for an interactive GIS query language. It is claimed that the extensions to SQL fail to provide a real GIS query language, and that they transfer to the programmer the burden of simulating spatial concepts.

Nevertheless, a considerable amount of effort has been directed towards overcoming these problems, with different extensions for relational systems. Proposed enhancements include the introducing of new spatial data types, functions, statements and graphical capabilities to represent spatial data. There is little concern with the output format, and most systems provide only default display screens.

Examples are the picture querying language of [59]; the SAND system [3], which extends a relational DBMS with spatial operations, keeping spatial data apart from textual data; the extended SQL of [41] which accepts geometric operators and where cursors can be defined to select geographic entities; and the work of [48].

5.2 Other textual query mechanisms

Various other (textual) query mechanisms are described in GIS literature. In [70] the query language QPF is discussed. This language is used in a vector-based GIS which incorporates the AI notion of frames to help users pose their queries. A query may be formulated using forms (visualization of frames) or by clicking menus. [61] extend the O2 object-oriented language with geometric and topological operators for a vector model, in cartographic applications. [40] uses a set of knowledge-based tools to perform queries for region classification.

5.3 Visual languages

A query language is said to be visual whenever the semantics of the query is expressed by a drawing. Most of these languages are based on graph visualization, often incorporating hypermedia facilities (e.g., [14]). Until now, visual languages and hypermedia concepts have not been much considered in the field of geographic information management and processing [10].

A visual GIS query language must allow selection of objects based upon their spatial location, and support reference to objects in a drawing, letting users directly manipulate representations of geographic phenomena. Some systems already allow users to specify fuzzy spatial concepts (e.g., nearness) through drawings, but visual querying facilities are still limited.

One important issue in GIS not yet supported by these languages is that qualitative analysis in planning is often based on visually comparing intermediate query results. Thus, visual queries must provide means for allowing users to directly manipulate alternatives, placing results in their proper graphical context. A good example of this last point is given in [19]: when the answer to a graphical query is a given city, it is useless to present a point in the screen, labelled by the city's name. Rather, the output must provide enough contextual (spatial) information so the user can effectively understand the answer.

5.4 Image databases

Many GIS applications require retrieval of images. Usually, images are preprocessed by the user according to some value partitioning scheme (e.g., using pixel classification strategies). Afterwards, queries retrieve images based on the classification result. In most GIS, images are treated as long (text) fields, and thus there is no possibility for content-based image retrieval (such as, for instance, described in [32]).

As stressed by [55], image databases must associate some sort of structure to images in order to allow content queries. Query results depend not only on the matching of image structure and query content but also on the accuracy of the image stored (due to the errors during the image entering and analysis process).

The spatial nature of image data is another issue that must be considered in queries on image databases. [29] discusses how to structure images for spatial querying purposes. As pointed out, most image databases cannot handle queries by image content, especially if accompanied by some spatial predicate (e.g., finding a river that has a 90' bend). Advantages and drawbacks of distinct approaches for querying images are discussed in [50].

6 Query optimization

As mentioned in [26], standard query optimization techniques are not always suited to scientific databases. It is remarked that many operations (e.g., those involving matrix transformations) are not easy to integrate into standard query processing systems, and suggested that in some cases optimization may be achieved by embedding part of this processing into the storage management subsystem.

Some of the problems are due to the amount, complexity and variety of data available. Other problems deal with the fact that strategies for accessing data on (approximate) spatial predicates are not the same as for other types of predicates. Furthermore, different users may want to analyze and cluster the same data in distinct ways, which complicates data placement strategies. Several of these optimization concerns are similar to those of optimizing object-oriented query languages [13, 9].

Spatial optimization proposals differ in capabilities and degrees of freedom they provide the optimizer, as a result of the manner in which spatial and non-spatial data is integrated [4]. Spatial query optimization usually involves two steps: filter and refinement [49, 37]. In the first step, a spatial index is used to select entities that may satisfy the query, and discard all others. The performance of this step depends on the spatial distribution of entities, as well as on the space decomposition strategy employed to build the index. In the refinement step, the query processor goes through the remaining candidates to select the desired result. The performance of this refinement step depends on the number of refined objects and their complexity.

[48] adopt the strategy of dividing the query into its spatial and non-spatial components, which are optimized and processed separately, and their results are afterwards composed. A similar policy is discussed in [4].

Related work concerns how to improve the efficiency of spatial operations through the use of computational geometry algorithms (e.g., [35]). As well, these operations can be speeded up by appropriate access methods (see next section).

7 Data Storage and Spatial Access Methods

There exists a great amount of literature on spatial data structures and access methods (e.g., [60]). According to [27], there is no consensus on which are the appropriate methods to implement into a DBMS, since there are few reports on performance of these methods.

To build an index, spatial entities are mapped into points in a k-dimensional space or circumscribed by containers (buckets, or rectangular boxes). These new entities (points or containers) are next placed in an index structure (e.g., trees, hash). The approaches to decompose space for indexing can be classified into [33]:

- bucketing data using the concept of minimum bounding rectangle, grouping nearby objects in hierarchies;
- dividing space into disjoint cells, which are mapped into buckets.

Reports on spatial index performance for GIS are usually related to restricted types of queries. Examples of comparative studies are [17] for different types of R-trees; [37], discussing query processing under several filter techniques; [33], for queries on line segment databases; and [58], where spatial join indices are considered to optimize spatial join operations on grid files. [17] differs from other studies in that the analysis considers that queries are interspersed with insertions and deletions to the spatial files. A benchmark for spatial queries was proposed in [65]. [36] analyze the influence of the container (box) shape to speed up the filtering stage.

8 Architectural Aspects

The rapid growth in GIS has resulted in a large number of systems, each of which with its own data handling and storage characteristics. Most GIS are still based on image and spatial data handlers coupled to a file manager, without any database facility. Data is partitioned according to the procedures that will be applied, and queries consist of sequences of procedures executed on these partitions.

The coupling of DBMS to GIS data processing requirements has been done according to three architectures:

- proprietary systems – a special-purpose database is tightly coupled with spatial data processing modules. Users cannot access the database directly and data cannot be migrated to other DBMS;
- layered systems – a standard DBMS is used as a basis for spatial data access functions. Users can access the database directly, and data can be ported into other systems. Most special purpose features (e.g., geometric and image processing modules) are implemented by external packages.
- extensible systems – these use the facilities provided by extensible relational or object-oriented DBMS embedding the spatial dimension in the system. The formulation of spatial queries is directly supported in the extensible query language.

Most of the above are implemented using relational systems. Recently, there have been some prototypes developed on top of object-oriented systems, but there is a lack of experimentation using real data. Examples of these architectures for different types of underlying DBMS are:

- *relational.* Data is usually in vector format. Storage management is often handled by two systems: the relational DBMS supports alphanumeric data, and another system processes spatial data [8]. One example is the ARC-INFO [46] commercial system, which combines thematic layers with the vector model.

- *extensible relational* – these rely on facilities provided by extensible relational models. Examples of the use of this approach are [30], who describe the modelling and implementation of GIS using Starburst; [67], who add R-tree and graphical display modules on top of POSTGRES; and [1]'s spatial database toolkit. The TIGRIS system [31], though claiming to be object-oriented, is in fact a modified relational system with handles.

- *object-oriented* – based on variations of the object-oriented paradigm, either using an object-oriented database or an object-oriented programming environment. Examples are the implementation of geographic relationships on the Zenith object manager [34]; the GODOT geo-object management system for vectorial data using ObjectStore [28]; the use of the O2 object-oriented system for cartographic applications [61] and environmental control [52]; and the comparative study reported on [45] using ONTOS.

- *rule-based* – these systems are designed with specific applications in mind, and usually result from coupling an expert system to a relational DBMS. Rules help users in their queries, by performing inferences on stored data. Most rule systems deal exclusively with alphanumeric data (e.g., [63]). Some support processing fuzzy queries on spatial relationships (e.g. [66, 40, 68]).

9 Other Issues

This paper discussed some of the issues that arise when dealing with database support for GIS, given the different points of view of end-users and database researchers.

There are many other (open) issues in database support for GIS applications. The first issue to consider is that different families of applications demand distinct types of database support: specific data models, analysis functions, storage and indexing schemes, special query languages and application-tailored I/O facilities. Thus, it seems unlikely that a general-purpose all-encompassing database for GIS will ever exist.

Second, one must keep in mind that many of these open problems are due to the nature of georeferenced data itself. Some of these problems are common to many other scientific database applications – e.g., the fact that there are large amounts of complex data, and that data varies with time. Other problems originate from the spatial dimension (which introduces for instance the issues of spatial integrity constraints and spatial query processing), errors in data collection and integration (see section 2), and output processing (e.g., different contexts and scales).

Third, some issues have been subject to database research, independent of GIS demands (e.g, active or temporal systems) and others are related to users' still incomplete understanding of their requirements (e.g., the formalization of spatial relationships).

Many other areas of database research, not mentioned in this paper, will contribute directly to the improvement of GIS. Among them can be cited: spatio-temporal management systems (e.g., [38, 64, 7, 39, 54]), data mining for determining correlation among stored data (e.g., [2]), active systems for decision support in planning applications [68, 71, 53], version control [8, 47, 44] long transaction management and recovery [6] and management of heterogeneous and distributed databases to integrate GIS data [20, 12, 22, 62, 16].

Acknowledgements

This research was partly financed by CNPq and FAPESP. The authors thank Gilberto Câmara for discussions on practical implementation problems.

References

[1] D. Abel. SIRO-DBMS: a Database Toolkit for Geographical Information Systems. *International Journal of Geographical Information Systems*, 3(2):103–116, 1989.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large

Databases. In *Proc ACM SIGMOD Conference*, pages 207–218, 1993.

[3] W. Aref and H. Samet. Extending a DBMS with Spatial Operations. In *Proc. 2nd Symposium Spatial Database Systems*, pages 299–317. Springer Verlag LNCS 525, 1991.

[4] W. Aref and H. Samet. Optimization Strategies for Spatial Query Processing. In *Proceedings 17th VLDB*, pages 81–90, 1991.

[5] S. Aronoff. *Geographic Information Systems*. WDL Publications, Canada, 1989.

[6] N. Barghouti and G. Kaiser. Concurrency control in advanced database applications. *ACM Computing Surveys*, 23(3):319–344, 1991.

[7] R. Barrera, A. Frank, and K. Al-Taha. Temporal Relations in Geographic Information Systems. *ACM Sigmod Record*, 20(3):85–93, 1991.

[8] P. Batty. Exploiting relational database technology in a GIS. *Computers and Geosciences: An international journal*, 18(4):453–462, 1992.

[9] J. Blakeley, W. McKenna, and G. Graefe. Experiences Building the Open OODB Optimizer. In *Proc ACM SIGMOD Conference*, pages 287–296, 1993.

[10] P Boursier and M Mainguenaud. Spatial Query Languages: Extended SQL vs Visual Languages vs Hypermaps . In *Proc 5th International Symposium on Spatial Data Handling*, pages 249–259, 1992. Volume 1.

[11] J. Carter. *Fundamentals of Geographic Information Systems: A Compendium*, chapter On Defining the Geographic Information System, pages 3–8. American Society for Photogrammetry and Remote Sensing, 1989.

[12] N. Chrisman. Deficiencies of Sheets and Tiles: Building Sheetless Databases. *International Journal of Geographical Information Systems*, 4(2):157–168, 1990.

[13] S. Cluet and C. Delobel. A General Framework for Optimization of Object-oriented Queries. In *Proc. ACM SIGMOD Conference*, pages 383–392, 1992.

[14] M. Consens, I. Cruz, and A. Mendelzon. Visualizing Queries and Querying Visualization. *ACM Sigmod Record*, 21(1):39–46, 1992.

[15] H. Couclelis. People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag LNCS 639, pages 65–77, 1992.

[16] D. Cowan, T. Grove, C. Mayfield, R. Newkirk, and D. Swayne. Managing Environmental Data - an Extension to the GIS Architecture. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pages 22–27, 1993.

[17] F. Cox. Análise de Métodos de Acesso a Dados Espaciais Aplicados a Sistemas Gerenciadores de Banco de Dados. Master's thesis, UNICAMP, december 1991.

[18] M. Egenhofer. Why not SQL! *International Journal of Geographical Information Systems*, 6(2):71–86, 1992.

[19] M. Egenhofer and A. Frank. Designing Object-oriented Query Languages for GIS: Human Interface Aspects. In *Proc 3rd International Symposium on Spatial Data Handling*, pages 79–98, 1988.

[20] A. Elmagarmid and C. Pu. Introduction to the special issue on heterogeneous databases. *ACM Computing Surveys*, 22(3):175–178, 1990.

[21] M. Goodchild et al. Integrating GIS and Spatial Data Analysis: Problems and Possibilities. *International Journal of Geographical Information Systems*, 6(5):407–424, 1992.

[22] R. Flowerdew. *Geographical Information Systems - volume I*, chapter Spatial Data Integration, pages 375–387. John Wiley and Sons, 1991.

[23] A. Frank and M. Goodchild. Two Perspectives on Geographical Data Modelling. Technical Report 90-11, National Center for Geographic Information and Analysis, 1990.

[24] M. Goodchild. Integrating GIS and Environmental Modeling at Global Scales. In *Proc GIS/LIS'91*, volume 1, pages 117–127, 1991.

[25] M. Goodchild. Spatial Analysis with GIS: Problems and Prospects. In *Proc GIS/LIS'91*, volume 1, pages 40–48, 1991.

[26] G. Graefe. Query Evaluation Techniques for Large Databases. *ACM Computing Surveys*, 25(2):73–170, 1993.

[27] O. Guenther and A. Buchman. Research Issues in Spatial Databases. *ACM Sigmod Record*, 19(4):61–68, 1990.

[28] O. Gunther and W-F Riekert. The Design of GODOT: an Object-oriented Geographic Information System. *IEEE Data Engineering Bulletin*, pages 4–9, september 1993.

[29] A. Gupta, T. Weymouth, and R. Jain. Semantic Queries with Pictures: the VIMSYS Model. In *Proceedings 17th VLDB*, pages 69–79, 1991.

[30] L. Haas and W. Cody. Exploiting Extensible DBMS in Integrated Geographic Information Systems . In *Proc. 2nd Symposium Spatial Database Systems*, pages 423–449. Springer Verlag LNCS 525, 1991.

[31] J. Herring. TIGRIS: A Data Model for an Object-oriented Geographic Information System. *Computers and Geosciences: An international journal*, 18(4):443–451, 1992.

[32] K. Hirata and T. Kato. Query by Visual Example. In *Proc. EDBT Conference*, pages 56–71, 1992.

[33] E. Hoel and H. Samet. A Qualitative Comparison Study of Data Structures for Large Line Segment Databases. In *Proc ACM SIGMOD Conference*, pages 205–214, 1992.

[34] Z. Kemp and R. Thearle. Modelling Relationships in Spatial Databases . In *Proc 5th International Symposium on Spatial Data Handling*, pages 313–322, 1992. Volume 1.

[35] H. Kriegel, T. Brinkhoff, and R. Schneider. The Combination of Spatial Access Methods and Computational Geometry in Geographic Database Systems. In *Proc 2nd Symposium Spatial Database Systems*, pages 5–22. Springer Verlag LNCS 525, 1991.

[36] H. Kriegel, T. Brinkhoff, and R. Schneider. Efficient Spatial Query Processing in Geographic Database Systems. *IEEE Data Engineering Bulletin*, pages 10–15, september 1993.

[37] H-P Kriegel. Query Processing in Spatial Database Systems . In *Proc. 2nd Symposium Spatial Database Systems*, pages 172–191. Springer Verlag LNCS 525, 1991.

[38] G. Langran. A Review of Temporal Database Research and its Use in GIS Applications. *International Journal of Geographical Information Systems*, 3(3):215–232, 1989.

[39] G. Langran. Issues of Implementing a Spatiotemporal System. *International Journal of Geographical Information Systems*, 7(4):305–314, 1993.

[40] Y. Leung and K. S. Leung. An Intelligent Expert System Shell for Knowledge-based Geographical Information Systems: 1 - the tools. *International Journal of Geographical Information Systems*, 7(1):189–200, 1993.

[41] R. Lorie. The Use of a Complex Object Language in Geographic Data Management. In *Proc. 2nd Symposium Spatial Database Systems*, pages 319–337. Springer Verlag LNCS 525, 1991.

[42] D. Maguire and E. Dangermond. *Geographical Information Systems - volume I*, chapter The Functionality of GIS, pages 319–335. John Wiley and Sons, 1991.

[43] D. J. Maguire, M. F. Goodchild, and D. W. Rhind, editors. *Geographical Information Systems - volume I.* John Wiley and Sons, 1991.

[44] C. B. Medeiros and G. Jomier. Managing Alternatives and Data Evolution in GIS. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pages 34–37, 1993.

[45] P. Milne, S. Milton, and J. Smith. Geographical Object-oriented Databases: a Case Study. *International Journal of Geographical Information Systems*, 7:39–56, 1993.

[46] S. Morehouse. The ARC/INFO geographic information system. *Computers and Geosciences: An international journal*, 18(4):435–443, 1992.

[47] R. Newell, D. Theriault, and M. Easterfieldy. Temporal GIS - modeling the evolution of spatial data in time. *Computers and Geosciences: An international journal*, 18(4):427–434, 1992.

[48] B. C. Ooi. *Efficient Query Processing in Geographic Information Systems.* Springer Verlag LNCS 407, 1990.

[49] J. Orenstein. Redundancy in Spatial Databases. In *Proc. ACM SIGMOD Conference*, pages 294–305, 1989.

[50] A. Hanson P. Suetens, P. Fua. Computational Strategies for Object Recognition. *ACM Computing Surveys*, 24(1):5–62, 1990.

[51] D. J. Peuquet. Toward the Definition and Use of Complex Spatial Relationships. In *Proc 3rd International Symposium on Spatial Data Handling*, pages 211–223, 1988.

[52] F. Pires, C. B. Medeiros, and A. Barros. Modelling Geographic Information Systems using an Object Oriented Framework. In *Proc XIII International Conference of the Chilean Computer Science Society*, pages 217–232, 1993.

[53] N. Pissinou, K. Makki, and E. Park. Towards the Design and Development of a New Architecture for Geographic Information Systems. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pages 565–573, 1993.

[54] S. Price. Modelling the Temporal Element in Land Information Systems. *International Journal of Geographical Information Systems*, 3(3):233–244, 1989.

[55] F. Rabitti and P. Savino. An Information Retrieval Approach for Image Databases . In *Proc 18th VLDB*, pages 574–584, october 1992.

[56] J. F. Raper and D. J. Maguire. Design Models and Functionality in GIS. *Computers and Geosciences*, 18(4):387–400, 1992.

[57] G-C. Roman. Formal Specification of Geographic Data Processing Requirements. *IEEE Transactions on Knowledge and Data Engineering*, 2(12):370–380, 1990.

[58] D. Rotem. Spatial Join Indices. In *Proc. IEEE Data Engineering Conference*, pages 500–509, 1991.

[59] N. Roussoupolos, C. Faloutsos, and T. Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, SE-14(5):639–650, 1988.

[60] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1989.

[61] M. Scholl and A. Voisard. *Building an Object-oriented System – the Story of O2*, chapter Geographic Applications – an Experience with O2. Morgan Kaufmann, California, 1992.

[62] I. Shepherd. *Geographical Information Systems - volume I*, chapter Information Integration and GIS, pages 337–360. John Wiley and Sons, 1991.

[63] A. Skidmore, P. Ryan, W. Dawes, D. Short, and E. O'Loughlin. Use of an Expert System to Map Forest Soils from a Geographical Information System. *International Journal of Geographical Information Systems*, 5(4):431–446, 1991.

[64] R. Snodgrass. Temporal Databases. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag LNCS 639, pages 22–63, 1992.

[65] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith. The Sequoia 2000 Benchmark. In *Proc ACM SIGMOD Conference*, pages 2–11, 1993.

[66] R. Subramanian and N. Adam. The Design and Implementation of an Expert Object-Oriented Geographic Information System. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pages 537–546, 1993.

[67] P. von Oosterom and T. Vijlbrief. Building a GIS on Top of the Open DBMS POSTGRES. In *Proc European GIS Conference*, 1991.

[68] C. Webster. Rule-based Spatial Search. *International Journal of Geographical Information Systems*, 4(3):241–260, 1990.

[69] M. Worboys, H. Hearnshaw, and D. Maguire. Object-oriented Data Modelling for Spatial Databases. *International Journal of Geographical Information Systems*, 4(4):369–384, 1990.

[70] J. Wu, T. Chen, and L. Yang. QPF: a Versatile Query Language for a Knowledge-based Geographical Information System. *International Journal of Geographical Information Systems*, 3(1):51–58, 1989.

[71] Z. Yang and D. Sharpe. Design of Buffer Zones for Conservation Areas and a Prototype Spatial Decision Support System (SDSS). In *Proc GIS/LIS'91*, volume 1, pages 60–70, 1991.