Spatio-temporal Database Systems: Foundations and Applications

Claudia Bauzer Medeiros
IC - UNICAMP Caixa Postal 6176
13081-970 Campinas, SP, Brasil
cmbm@ic.unicamp.br

Abstract

Este texto analisa problemas de pesquisa em bancos de dados espaço-temporais segundo seu gerenciamento por um sistema gerenciador de bancos de dados (SGBD). A análise do termo "bancos de dados espaço-temporais" permite deduzir sua definição de forma apropriada: são sistemas de bancos de dados que gerenciam dados que variam no espaço e no tempo. Dada esta caracterização, qualquer estudo destes sistemas exige combinar conhecimento em sistemas gerenciadores de bancos de dados, bancos de dados espaciais e bancos de dados temporais. Além disso, é importante ter um conhecimento básico dos dados manipulados e de problemas relativos à sua modelagem, face às aplicações em que são usados. O texto trata todos estes aspectos - SGBDS espaciais, SGBDS temporais e sua combinação – aliados às questões da natureza espaço-temporal dos dados, restringindo-se a sistemas que manipulam dados geográficos. Outros tipos de dados espaço-temporais – por exemplo, de astronomia ou medicina – fogem ao escopo deste trabalho.

This text analyzes research problems on spatio-temporal databases, from a database management system DBMS viewpoint. The parsing of the term "spatio-temporal databases" allows generating an appropriate definition thereof: they are DBMS that manage data which vary in space and time. Given this characterization, any study of these systems demands combining knowledge in DBMS, spatial databases and temporal databases. Moreover, it is important to take into account a basic knowledge of data being manipulated, as well as issues concerning their modeling, given the applications where they are used. This text considers all these aspects - spatial and temporal DBMS and their combination - allied to considerations on the spatio-temporal nature of the data, restricted to geographic data. Other types of spatio-temporal data - e.g., in astronomy or medicine - are outside the scope of this work.

Keywords

Spatial databases, geographic information systems, temporal databases

I. Introduction and historical background

THE notions of space and time are omnipresent in our daily life. Nothing more natural, therefore, than to develop systems to help us handle these issues. Unfortunately, spatial data is very voluminous; if, moreover, one considers their variation in time, this volume grows exponentially. Hardware limitations prevented, until recently, the development of database management systems (DBMS) for spatio-temporal data. It is only in the 90's that storage device technology evolved to allow storing and accessing larger data sets, thus setting the infrastructure for research and development in the domain of spatio-temporal DBMS.

This trend was preceded, in the 80's, by the appearance of strictly temporal database management systems [1], generating hundreds of papers – e.g., see the bibliography of [2]. Temporal DBMS concern the handling of data that vary with time, but where the notion of space is ignored. The main motivation for this research direction was given by business and administrative applications. Stock and sales control are the most common applications of the first kind, whereas personnel management concern the second kind. In the mid-80's, temporal database systems started to be used in scientific applications. This gave origin, in turn, to a new research field in computer science, that of *scientific databases* [3].

A large group of scientific applications were concerned with the management of time series (e.g., in statistical studies), where the novelty resided in the fact that data were now managed by DBMS instead of being stored in unstructured flat files. In the natural sciences, temporal databases started being used to store environmental-related measurements and observations (rainfall, temperature and so on) but still in a limited way. More often than not, computer systems dealing with environmental temporal-varying data did not rely on database systems, and were concerned with algorithmic and simulation aspects, disregarding issues concerning efficient data storage and management. Thus, most of the research on temporal databases remained restricted to business-oriented applications.

This scenario was drastically changed in the beginning of the 90's, with the consolidation of the field of spatial DBMS. Here, the emphasis was on storing data about space, motivated by geographic studies (e.g., [4], [5], [6], [7], [8]). Research on geographic information systems (GIS), which up to then had been restricted to end-user applications, was recognized by computer scientists as a worthwhile field. A GIS from then on would denote any kind of information system that manipulated maps and/or remote sensing data. Nevertheless, spatial data was treated from a static viewpoint, and time evolution was not considered.

Thus, from a data management perspective, two research and experimentation lines were being pursued – one concerning temporal evolution, regardless of space, and the other spatial characteristics, for a fixed moment in time. Spatio-temporal evolution was not supported by any system, and it was up to end users of spatial databases and of GIS to build up additional software to consider the changing of spatio-temporal features along time. From another viewpoint, the survey presented in [9] postulates that the role of spatio-temporal systems is in the tracing of the lineage of spatial objects and their properties. They indicate as basic functions those of *inventory, analysis, scheduling and quality control.*

Spatio-temporal data management is still very much at the experimentation and research level. End-users still work with nontemporal spatial data files. They basically generate static views of spatial data, for different points in time, and compare these views manually or visually in order to figure out temporal evolution. Sometimes, this may be helped by an underlying versioning mechanism [10].

Some limited proposals to jointly manage space and time appeared in the mid-90's, for environmental problems (e.g., [11], [12], [13]). This is a field that is receiving growing attention from computer scientists, motivated by global scale problems – such as global warming, pollution and biodiversity preservation. At another scale, economic motivations are pushing forward the development of spatio-temporal systems, notably in transportation and agriculture.

Nevertheless, in the beginning of the 21st century, there is no satisfactory solution to the problem of managing spatio-temporal data. There are several questions that must be tackled, covering a spectrum that starts at the data sampling and collection end and finishes with data storage and communication concerns. It is true that solutions exist for precise problems, and in many cases one can find software that has been custom-built to deal with a givenissue, for specific data sets. Examples are systems that manage data concerning water or air pollution ([14], [15]), developed to store these data and use them in distinct kinds of simulations.

There is not, however, any general purpose spatio-temporal database system, as opposed to the wide range of generic DBMS for standard (e.g., table-based) data sets. This paper presents several kinds of problems that exist to develop such generic systems. Sections II and III present the basic framework which will be used to discuss these issues. The four subsequent sections analyze open problems under this framework. Finally, section VIII points out future trends in the field.

II. Basic framework

A. Characterizing spatio-temporal data

SPATIO-TEMPORAL DBMS are first and foremost database systems that manage a specific kind of data – those that reflect phenomena that vary in space and in time. In order to discuss these systems, one must thus consider two factors: the basic properties of database management systems and the characteristics intrinsic to spatio-temporal data.

Database management technology dates back to the 70's and has been consolidated by several commercial products. These DBMS provide a stable set of functions that allow concurrent use of massive data sets. Spatio-temporal data introduce a set of perturbations into this scenario, due to three properties – volume, intrinsic heterogeneity and functions to be applied to these data.

There are distinct kinds of spatio-temporal data sets. For instance, those used in the design of engineering artifacts are spatial in the sense that they must cater to 3D features, and temporal to accommodate the design evolution with multiple alternatives. By the same token, some types of video and medical image databases, as well as astronomy databases, can also be considered in the spatio-temporal context. This paper deals with a specific spatio-temporal family – geographic data, also called *georeferenced*.

Geographic data portray any feature that is related to the Earth. The term *georeferenced* refers to data about geographic phenomena associated with its location, spatially referenced to the Earth [16]. Thus, one can say that "the Titicaca lake", or "the Empire State Building" relate to geographic data, because they refer to identifiable geographic entities, whereas "John Doe"is not usually seen in a geographic context.

These entities are not georeferenced until coordinates have been assigned to them, under some projection system. If "John Doe" corresponds to a database record which contains an "address" field (e.g., residence, place of work), then this record can be georeferenced if the address is associated to geographic coordinates. This text treats the terms "geographic" and "georeferenced" as synonims, in spite of this difference between the two concepts, and assumes that data on geographic entities have been properly georeferenced.

A georeferenced entity is usually considered according to two main components: a geographic characterization, also called spatial attributes (including location and geometric specification) and conventional (descriptive) attributes, also called features or non-spatial attributes. For instance, in 2D, the lake's spatial extent is approximated by a sequence of points along its margins; and its remaining characteristics – average water temperature, pH and so on – are stored as additional non-spatial attributes. The same applies to man-made entities – e.g., the building is represented by a polygon (its 2D projection on the ground) and non-spatial attributes concern number of offices, floors, elevators, average number of visitors per day and so on. In 3D, the building's height is represented as an additional attribute, whereas the lake may be represented by a set of isolines that portray the terrain under the water. For simplicity sake, this paper will deal only with 2D representations, assuming that the z dimension does not vary with time.

The database containing these data on the lake or the building is a spatial (geographic) database, but it is not yet spatio-temporal. The introduction of the time dimension requires storing along another set of attributes, to describe time. For instance, the lake's average temperature and pH may vary with seasons (and thus, for the same spatial description, several records will have to be stored, each of which "timestamped" with a distinct season name). An even more complicated situation arises if the lake's geometry changes with time. These may be permanent – due to changes along the lake's border, e.g., due to pollution or urban development – or temporary, e.g., due to floods or draught. Again, many more records will have to be stored concerning this spatial evolution. The same applies to the building, that may change with time – more offices/elevators, but also a new wing.

There are several other ways to consider spatio-temporal evolution. One example is the classification of changes introduced in [9], where the term includes both the notion of modification as well as movement. Semantic changes correspond to variations in attributes over time, for a static spatial distribution of a geographic phenomenon (i.e., only nonspatial attributes change value); spatial changes correspond to modifications in the spatial attributes, and they may be static, *when two snapshots are compared* (sic) or transitional, when states of an event are compared at different sites. Finally, temporal changes are either spatially fixed or involve movement (e.g., in mobile applications).

B. Characterizing the temporal dimension

SPATIO-TEMPORAL data for a geographic entity can be characterized by the following:

- A basic description, indicating its spatial and non-spatial attributes to be stored. In database theory, this is called the *schema* of the data;
- As many records with this schema as there are variations on attribute values along time. These variations may concern spatial and non-spatial attributes alike. Each such record has a set of timestamps associated to characterize the time for which these values are considered. In a database, this means additional fields to store time, and which are incorporated to the schema.

There still remains the issue of the timestamps themselves. What is their granularity, and what do they stand for? This question is examined by research on temporal databases, where three basic kinds of timestamp are considered – *valid, transaction* and *user-defined* time. Valid time concerns the temporal description for which the attribute values were valid – e.g., if the building had a new annex built in 1954, then the valid time value stored is "1954". Transaction time marks when the values were stored in the database – e.g., if the data on the building annex was only stored in 1998, then there will be one valid timestamp "1954" and on transaction timestamp "1998". User-defined time is based on user dependent time referentials. For the building, suppose users measure time in zoning legislation changes; then, there would be a third time attribute that would correlate changes in time to legislation modifications. User-defined time is generally discarded for automatic temporal DBMS management.

The pair < valid time, transaction time> timestamp is normally used in (bi)temporal databases to store distinct valid versions of the world "as-of" the transaction timestamps. In geographic applications, the transaction timestamp is normally ignored. Again, this originates from end-user requirements and work habits. As mentioned in section I, users of geographic information systems have only recently learned to rely on DBMS. Furthermore, temporal DBMS are still a matter of research. Thus, end-users still restrict themselves to working with valid time notions, which suffice for several kinds of simulations and decision making. However, transaction time introduces another facility - it allows "navigating along the time axis" (in temporal database terms, rollback or roll forward operations) to perform simulations using past database states. Last but not least, it allows recording forecasts of the future (future-valued valid timestamps) as imagined today (present time transaction timestamps), which will in the future allow studing the evolution of knowledge. This kind of double timestamping is not considered in geographic applications and is left for future research; this paper will only consider valid time.

Even if the kind of timestamp is already decided on, there is still the need for discussing level of timestamping, time granularity and interpolation functions. This decision has repercussions on storage management and query processing. The *level of timestamping* concerns the granularity with which users want to mark changes in time. There are basically two levels: attribute-based and record-based. In record-based stamping, every time any attribute changes value, the whole record is duplicated, the value is modified and there is just one timestamp for the whole record. In attribute-based timestamping, there are time fields associated with each attribute that can change with time. When an attribute changes, instead of duplicating the record, only the attribute is duplicated, its value modified, and the associated timestamp recorded. Each solution presents its pros and cons. Hybrid solutions contemplate both kinds of timestamping, simultaneously. For the purpose of this paper, this decision is immaterial. Attribute-level timestamp will be assumed, because it facilitates explaining some concepts.

Time *granularity* deals with the kind of time unit used for expressing timestamps - minutes, seconds, hours, seasons; and does one record intants or intervals? Again, this depends on user needs. There must be a compromise between a granularity that is too fine and another that is too coarse. Furthermore, since distinct entities vary with different periodicity, there remains the problem of what to store to allow comparing several spatio-temporal objects. In temporal database theory, one uses the concept of *chronon* to denote an atomic time unit to be used as a basis to measure time. Frequent chronon values are seconds, miliseconds or minutes. Again this is immaterial in this paper, but one must not forget this issue, since it has implications in data integration and interoperability.

Finally, there is the matter of the *functions* to be used to derive attribute values between two timestamps in which these values were recorded. These interpolation functions are usually implicit and assumed by the DBMS – step and linear functions are two examples. In geographic applications, the stepwise function is the most common - i.e., an attribute's value is assumed constant from the time it is sampled to the next time this value is recorded. This can also be complemented by alternative timelines – e.g., continuous, cyclic, branching or terminating, which enhances the options database designers will have to model spatio-temporal phenomena.

C. Characterizing a DBMS

GIVEN the nature of georeferenced data, present DBMS facilities need to be extended to provide adequate support to spatio-temporal applications. This paper will analyze a few of these needs under a simplified framework.

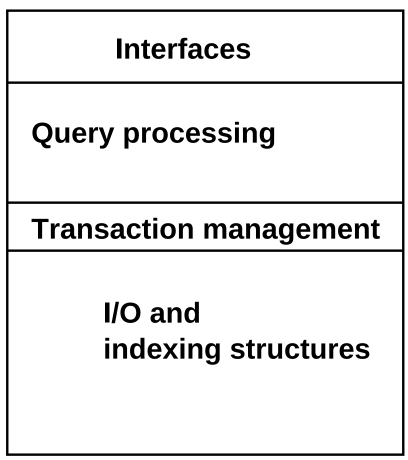

Fig. 1. DBMS schematic characterization - layers width show relative weight of spatial and/or temporal research

A DBMS can be schematically portrayed in a four-layer architecture, as shown in figure 1: interface, query processing and optimization, transaction management and I/O-buffer handling. The figure's layers are spatially organized (!) to show the amount of work done in each of these facets for spatial and/or temporal databases. The top layer - interfaces - is responsible for offering to the user adequate tools to access the underlying data. There are basically two kinds of facilities considered at this level: query languages and API (application programming interfaces). There are furthermore additional tools that help design the database.

This is followed by the query processing layer, which optimizes user requests formulated at the interface level and generates code to be executed at the lower levels. Transaction management layer is responsible for ensuring concurrency control and recovery. Finally, the I/O handling layer is reponsible for managing data transfers from secondary storage. Traditionally, indexing structures are also considered at this level.

This section set up part of the basic framework for analyzing spatio-temporal DBMS: data structures, the temporal dimension and a simplified DBMS characterization. It remains to introduce the missing dimension – spatiality – discussed next.

III. Geographic data

GEOGRAPHIC data are time and space specific and come in distinct formats, from different sources and geographic locations, and are captured by various types of devices[1]. As pointed out in [17], database specialists often ignore many concerns that underly end-users' geographic applications, such as data accuracy evaluation, feature generalization, fuzziness of geometric boundaries or need for versatile data analysis tools. As a consequence, many proposals to manage spatio-temporal data fall short of end-user needs. End-users, on the other hand, deal with data capturing and sampling, as well as in interpolation and modeling functions, but ignore the need for proper data management procedures.

Applications involving geographic (spatio-temporal) data vary widely, and so do user requirements. Examples of such applications are those dealing with urban planning, route optimization, public utility network management, demography, cartography, agriculture, natural resources administration, coastal monitoring, fire and epidemics control [18]. Each type of application deals with different features and geographic and temporal scales and properties.

Data sources are also varied. In the 80's, they were more restricted – census data, some photos and digitalized maps. Nowadays, remote sensing data are the primary source for environmental geographic applications, whereas urban cartography, utility needs and census profiles are sources for urban planning applications. This introduces yet another factor in data management – the speed with which data is captured by sensors allows fast collection of massive data volumes. There is a growing set of applications where data are collected and geo-referenced directly using a GPS (e.g., [19]).

Though more and more data are available, they cannot be queried. Indeed, raw sensor-provided data must be pre-processed and integrated before they can be used. Up to now, there are no satisfactory means of processing general remote sensing data to allow real time database entering and querying. Real time updates followed by queries are by no means a problem in remote sensing only. In the business world, this is also a complicated issue, since there is still no overall satisfactory solution to the problem of updating commercial data warehouses (e.g., sales records). Furthermore, big commercial chains are now trying to cope with this issue in order to better plan the organization of the supply and demand chain. If it is a challenge to manage these kinds of data, it is much more of a problem in the case of geographic data, which must always undergo a wide spectrum of transformations before they can be useful in geographic applications.

Besides the issues of data collection and sampling procedures, one must also consider the ways end-users view the real world. This directly influences spatio-temporal modeling and therefore the way queries are posed and processed. The modeling of georeferenced data by end-users is associated with different perceptions of the world: the *field* model and the *object* model [20], [21]. 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" [22].

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

The *object* view treats the world as a surface littered with recognizable objects, which exist independent of any definition (e.g., the Empire State building or the Titicaca lake). In this model, two

[1] The text contained in this section is basically a copy of [17]

objects can occupy the same place (e.g., the building on a street or a beach on the lake's margins). 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). In this case, coordinates are not stored, but rather derived by the position of the cells in the scan order. 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.

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

All spatio-temporal DBMS rely on the vector format. The reason is that one wants to record time changes in spatial attributes, which are indirectly associated with geometric properties. In the tesseral format, geometry is fuzzy and depends on the classification of values imposed on the data. For instance, given a satellite image of an agriculture area, the specification of plot limits and crops may vary widely depending on the image resolution and human expert classification. Clusters of trees or clouds may block part of the image. If successive images are taken of this same area (say, in one week intervals), then the valid timestamps are associated with the whole image and not with individual entities that may be derived from it. This complicates any spatio-temporal analysis on these data. Thus, usually, tesseral data are first vectorized (e.g., by image processing techniques) to extract meaningful objects, and only afterwards are timestamps associated. For this reason, this paper only considers vector data as liable to temporal management

Storage structures directly influence index structures as well as query processing and optimization. The rest of this paper considers spatio-temporal data under the DBMS framework introduced in section II-C.

IV. Interface issues

THERE is still very little on spatio-temporal database interface issues. This section examines some problems in interfaces of spatial systems and for temporal databases.

Temporal database interfaces are usually concerned with showing data aggregated according to time constraints, in tabular forms. Another possibility is introducing directed graphs to portray evolution along time intervals. Finally, animation has recently started being considered a means of showing data temporal evolution.

Georeferenced data are much more difficult to present and to manipulate, requiring a combination of graphical and textual representations and complex transformations of data from the GIS geographic database to the user and vice-versa [23], [24]. In particular, the problem of interface building is completely ignored and application designers have to hardwire the interface design directly into the geographic application code.

Figure 2, taken from [25], characterizes research in interfaces for GIS. Research on human factors is based on presenting concepts according to the user's mental framework. Research on languages is geared towards translating concepts from the user level to an intermediate (interface level) language L_e, and from this intermediate level to the internal database language L_i. In analogy to relational DBMS, the internal language L_i would correspond to the relational algebra while the external language L_e could be compared to SQL. Research on interface architectures is geared towards specifying building blocks and functions to construct interfaces.

Human factors research is heavily based on cognitive studies, which play a very important role

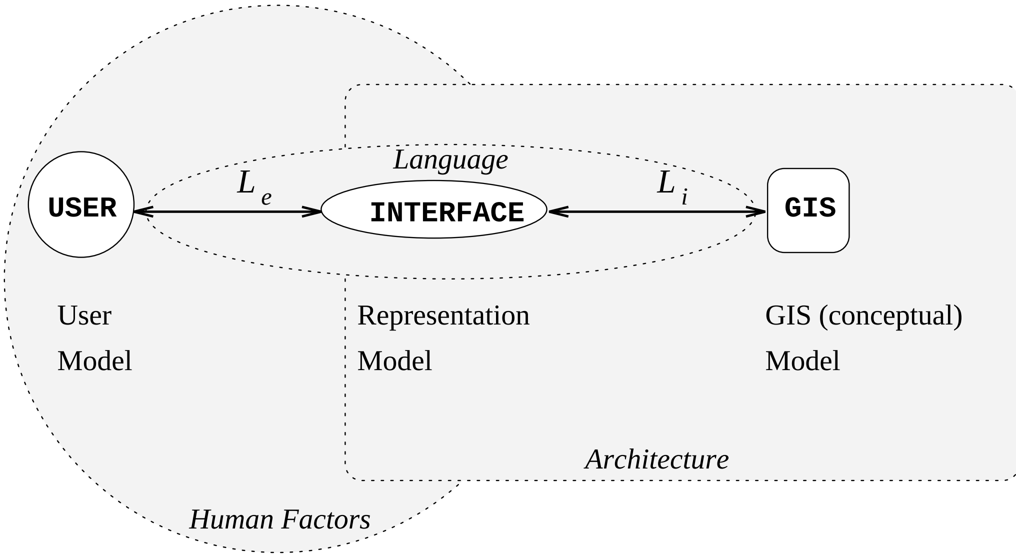

Fig. 2. Research directions in interfaces for geographic systems

in application interfaces [26]. The same data elements may have to be presented in several ways, depending on the application or even in the user's profile. For geographic applications, this is also related to exploring cartography presentation variables such as size, color, texture, grain and form. As shown by Monmonier's classic work "How to lie with maps" [27], these variables can be manipulated to distort reality.

Research at the language level is geared mostly towards definition of external languages L_e [28], [29]. These are used to convey the interaction of the user with the interface representation model. Papers dealing with interface architectures try to to build an interface representation model that can be efficiently mapped to and from spatial data models, e.g., [30], [31], [32]. They also define different type of internal interface organizations which include modules for cartographic presentation and customization, as well as some means of processing the internal language L_i.

Spatio-temporal systems that use animation are usually restricted computer graphics environments, showing movement of 3D objects in space. In the geographic context, there is a limited kind of spatio-temporal display in applications involving traffic simulation, and walking through virtual cities [33]. Most other situations of spatio-temporal evolution are not based on the use of DBMS, but rely on combining sophisticated simulation functions with computer graphics. This corresponds to the field of scientific visualization, which still needs to be better integrated to spatio-temporal DBMS.

V. Query processing

A. Query classes

TEMPORAL queries already have standards (e.g., TSQL2 [34]), but there is still no consensus on geographic query language constructs. The queries to be posed from a geographic DBMS can be roughly classified into what [12] calls the W-triangle:

- *What* kind of data are in this region? (spatial predicate, non-spatial answer)
- *Where* do these data refer to? (nonspatial predicate, spatial answer)
- *When* do these data refer to? (temporal valid time query)

These three questions reflect the spatio-temporal nature of georeferenced data.

A fourth question – How *accurate* is it? – 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. Yet another set of questions refer to the fact that a large set of applications involve urban or environmental planning and decision support. Thus, different users may pose the same query but will want to be shown answers tailored to their level of need. Queries at this level are not posed by the user, but are inferred by the system from the user's profile: who is posing the question, what kind of application domain is this query involved in, what is the space-time scale and granularity considered. Whereas for the first four questions some work has already been started, the other profile-based queries are still in their infancy and are treated mostly in AI, but not in the geographic context.

B. Query components

As stressed in [35], in analyzing geographic queries, one can use as a basis the standard SQL query framework of

```
SELECT result
FROM data sources
WHERE set of predicates
```

in the sense that every query is processed by imposing retrieval predicates on a set of data sources, and return the results organized according to some pre-defined template/set of attributes. This template is then presented to the user in a multitude of ways. In this kind of analysis, it is very important to dissociate the result (query processing phase) from its presentation (an interface issue). Result schemata were discussed in section II, data sources in section III and presentation in section IV. The rest of this section concentrates on predicate analysis and query optimization.

One big difficulty is the definition of something equivalent to relational algebra, called by some researchers *map algebra* [36]. Little has been done to identify and formalize spatio-temporal key functional requirements, in terms of basic spatial operators and supported relationships, and available systems lack sophisticated analysis facilities [37]. 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 spatio-temporal DBMS should provide [38]: 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 demand sophisticated data analysis functions; the latter are the focus of the *spatial analysis* view.

[39] distinguishes between the following classes of spatial relationships:

- topological relations - those that are invariant under topological transformations like translation, scaling or rotation. Examples are *adjacent, inside, disjoint.*
- direction relations - those that establish relative positioning of elements within some positioning system (e.g., *north, south*).
- metric relations - those that can be expressed in scalar form defining measurement values (e.g., *distance*).

Spatial relationships and operators depend on factors such as scale, time, point of view and preciseness of their specification (e.g., [40], [41]). [42] 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 systems 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 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. Examples of spatial query textual languages include GeoQL [43] and PicQuery [44]. An example of an implementation of a query subsystem for an object-oriented DBMS is the work of [45], [46]. More recently, visual languages (e.g., [47], [48]) started being proposed for spatio-temporal systems.

C. Query optimization issues

AS mentioned by [49], standard query optimization techniques are not always suited to scientific databases. It is remarked that many operations (e.g., those involving matrix transformations – in the case of tesseral data) 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 spatial predicates, which are often fuzzy, 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. Even though clustering and placement are not new problems, here the issue is the size of the data records, because of the length of spatial attributes. Moreover, geographic entities are embedded in each other, and thus optimizing clustering strategies is a problem similar to that of defining clustering of objects in object-oriented databases, which has not yet been solved.

Spatial query optimization usually involves two steps: filter and refinement [50], [51], [52]. In the first step, a spatial index is used to select entities that may satisfy the query, and discard all others – see section VII. 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.

A different kind of processing solution is given by [10], using the notion of versions in databases, but without considering performance. This solution is based on the DBV database versioning mechanism which allows alternative database versions to coexist within the same data space. This enables spatio-temporal queries for analysis of temporal evolution, both within one database state and across several temporal states, each of which depicting one spatial database. Queries in this case are handled in three steps: first, one selects all database states that fall within the desired time frame; next, standard spatial queries are run on each of these databases; finally, comparisons and simulations are performed on the results of these queries.

VI. Transaction management

THERE is virtually nothing in the literature concerning transaction management for spatio-temporal systems (or even for temporal databases, for that matter). The closest one can get to this topic is when transactions act on multidimensional data [53], but without a specific application domain. In fact, spatio-temporal applications and, more specifically, geographic applications, usually involve the so-called *long transaction* management mechanism – e.g., [54].

Therefore, the tacit assumption in all spatio-temporal DBMS is that the appropriate transaction model is the one that deals with long transactions, using large amounts of data. In such a scenario, concurrency is very limited, and more often than not handled by an appropriate data check-in/check-out mechanism [55]. This is often the case in geographic planning applications. The relevant data on the region under study are "checked-out" by the people involved in the planning activity. These data are then processed and analyzed in a separate work space, for periods that may last from one day to several months. Next, the updated scenario is returned ("checked-in") to the database, and additional integrity checks must be performed to allow its reintegration. This kind of mechanism is often associated with version stamping and version control, to allow data recovery as-of a certain time frame.

The other situation concerns online updates of spatio-temporal data. These are handled by the DBMS as any database update, and standard transaction management is used. Again, because of the object embedding property mentioned in the previous section, locking mechanisms borrow from object-oriented concurrency control mechanisms. There are, however, situations where this is not sufficient

- e.g., in risk management [56], which requires real time data handling. In such a case, several DBMS functions are eliminated, and special purpose systems are built to filter input data, integrate them to existing data, analyze the result and return it to the user. Sophisticated cartographic displays are abandoned in favor of less maps with less details or 3D graphs. This type of situation is handled more adequately by active databases, where specific states are monitored and triggers are fired when a given situation occurs [57]. There is, however, very little research that combines active databases with management of spatio-temporal data, and dynamic cartographic displays. Work in this area is only beginning, and concurrency or recovery are not considered.

VII. I/O and indexing

THERE exists a great amount of literature on spatial data structures and access methods (e.g., [58], [59]), as portrayed in figure 1. There is however very little on spatio-temporal structures. Usually, they consist of either appending timestamps to spatial data structures or, alternatively, appending spatial attributes to temporal data structures.

Temporal I/O and indexing is treated under different guises, but mostly using hierarchical structures, often based on B-trees and their variations. Several structures are based on the premises presented in [60], which is concerned with reducing redundancy and keeping recent time data in memory and older data in persistent storage. The idea behind this reasoning is that users are usually want faster access to recent records, and tolerate slower access to older (historical) data. The basic structure described in the paper is the Time-Split B-tree, which is based on indexing data on time periods in such a way that splits preserve time ordering. Splitting on time T keeps entries with timestamp smaller than T in the old node, the remaining entries go to the new node, which also must contain the version valid at the split time.

Another kind of structure involves adapting B+-trees to index one-dimensional time databases [61], where each tree leaf indexes a given valid time database. This structure partitions the database according to the timestamps; each set of values associated to a timestamp is an atemporal database, with its own index. Then, the set of atemporal databases and indices is indexed on valid time by the B+-tree. Other temporal indexing mechanisms are discussed in, for instance, [62], [61].

To build a spatial 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 [63]:

- 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 [64] for different types of R-trees; [51], discussing query processing under several filter techniques; [63], for queries on line segment databases; and [65], where spatial join indices are considered to optimize spatial join operations on grid files. [64] differs from other studies in that the analysis considers that queries are interspersed with insertions and deletions to the spatial files. A pioneer benchmark for spatial queries was proposed in [66]. [67] analyze the influence of the container (box) shape to speed up the filtering stage. Spatial access methods have also been tested and compared in [64], [68], [69], [70].

Spatial join operations present a very complicated issue in spatial query processing, and thus merit specific mentioning in any paper on spatial DBMS. Join predicates involve combining two sets of spatial objects according to their overlapping or proximity. Examples of papers dedicated to this issue, for distinct kinds of data and applications, are [59], [71], [72].

R-trees and quadtrees are examples of multidimensional access methods. A comprehensive comparison of such access methods is the review presented by Gunther and Gaede [73]. More recently, a new kind of multidimensional indexing mechanism has appeared, originated by the M-tree family of structures. This allows creating indices for data sets in metric spaces, to answer queries on dataset similarities. Each object is represented by an m-dimensional characteristic vector, which is stored in the tree. This is finding acceptance in several applications where searchcriteria are based on multiple relationships among stored objects, and query predicates imposed on distinct kinds of metrics – e.g., image databases [74], [75]. Yet another family of metric access methods (Omni-methods) has been proposed by [76], which also presents a good comparison of multidimensional access methods.

Finally, spatio-temporal indexing is now attracting attention. An early example of modification of R-trees with time is [77]. [78] uses versioning of quadtrees to keep track of changes in images through time. Recent research on spatio-temporal indexing and dataset generation is now appearing in the context of moving object databases, but can also be sometimes used for non-moving object applications. Examples are [79], [80], [81]. See conclusions for more pointers on moving object literature.

VIII. Conclusions and future trends

THIS paper discussed several issues concerning the management of spatio-temporal geographic data management, from a database point of view. It was pointed out that the problems arise both from the nature of the data and application domains, but also from inadequacies of present DBMS to deal with performance and functional requests.

Many other problems remain to be solved in the context of spatio-temporal data, which were not mentioned in the paper. Some of these problems include:

- Modeling and design of spatio-temporal applications and their transformation into database entities. This falls into an area that is increasingly combining software engineering techniques with database design. Examples are [82], [83], [84], [85], [86], [36], [87], [88], [89].
- Management of moving objects - the so-called mobile applications. Movement data introduce yet another dimension to these questions, since there is the problem of keeping track of position along time, both in terms of sampling and storage techniques as well as data interpolation for querying. Examples of research in this direction are [90], [91], [92].
- Warehouses and interoperability - besides business warehouses, there is now a concern with building warehouses of spatio-temporal data, for global environmental applications. Associated questions involve integration and interoperation of geographic data and software. Examples of research in these directions are [93], [94], [95], [96], [97], [98], [91].
- Dynamic modeling and representation of natural processes and their management within a DBMS framework. Examples are [99], [100], [14].
- Data quality standards and metadata concerns - the entire reliability of a system is based on the quality of its data. Research in this area is often tackled by metadata standards, that refer to a "quality" field. This is, however, not sufficient and there is still much to be done. Examples of research in this area are [101], [102], [103], [104], [105], [106].

Acknowledgements

The work reported in this paper was partially financed by CNPq, and FAPESP, and by the projects PRONEX MCT/FINEP SAI (Advanced Information Systems) and CNPq/NSF (Interoperability in GIS).

References

[1] R. Snodgrass, "Research concerning time in database, project summaries," *Sigmod Record*, vol. 15, pp. 19–39, December 86.
[2] M. Soo, "Bibliography on Temporal Databases," *ACM SIGMOD RECORD*, vol. 20, no. 1, pp. 14–23, 1991.
[3] J. French, A. Jones, and J. Pfalz, "Summary of the Final Report of the NSF Workshop on Scientific Database Management," *ACM Sigmod Record*, vol. 19, no. 4, pp. 32–40, 1990.
[4] E. Apolloni, F. Arcieri, L. Barella, and M. Talamo, "Requirements and Design Issues of Spatial Data Handling Systems," in *Advances in Database Systems: Implementations and Applications*, pp. 49–68, Springer Verlag Courses and Lectures 347, 1994.

[5] F. Arcieri, E. Apolloni, L. Barella, and M. Talamo, "Hermes - an Integrated Approach to Modelling Data Base Systems Design," in *Advances in Database Systems: Implementations and Applications*, pp. 71–93, Springer Verlag Courses and Lectures 347, 1994.
[6] M. Casanova, A. Hemerly, and A. Furtado, "Cooperative Environments for Geographic Databases: a Prescriptive Analysis," in *Anais, 8 Simpósio Brasileiro de Bancos de Dados*, pp. 266–279, 1993.
[7] 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*, pp. 22–27, 1993.
[8] K. Fedra, "GIS and Environmental Modelling," in *Environmental Modelling with GIS* (M. Goodchild, B. Parks, and L. Steyaert, eds.), pp. 35–47, Oxford University Press, 1993.
[9] T. Abraham and J. Roddick, "Survey of Spatio-temporal Databases," *GeoInformatica*, vol. 3, no. 1, pp. 61–99, 1999.
[10] C. B. Medeiros and G. Jomier, "Using Versions in GIS," in *Proc. International DEXA Conference*, pp. 465–474, 1994. Springer Verlag Lecture Notes in Computer Science 856.
[11] A. Frank, I. Campari, and U. Formentini, eds., *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Lecture Notes in Computer Science 639, Springer-Verlag, 1992.
[12] G. Langran, "Issues of Implementing a Spatiotemporal System," *Intl. Journal of Geographical Information Systems*, vol. 7, no. 4, pp. 305–314, 1993.
[13] P. Story and M. Worboys, "A Design Support Environment for Spatio-Temporal Database Applications," in *Proc COSIT*, Springer Verlag Lecture Notes in Computer Science 988, pp. 413–430, 1995.
[14] T. Clemen, "The use of Scale Information for Integrating Simulation Models into Environmental Information Systems," *Ecological Modeling*, vol. 108, pp. 107–113, 1998.
[15] D. Sui and R. Maggio, "Integrating GIS with Hydrological Modeling: practices, Problems, and Prospects," *Computers, Environment and Urban Systems*, vol. 23, no. 1, pp. 33–51, 1999.
[16] J. Carter, *Fundamentals of Geographic Information Systems: A Compendium*, ch. On Defining the Geographic Information System, pp. 3–8. American Society for Photogrammetry and Remote Sensing, 1989.
[17] C. B. Medeiros and F. Pires, "Databases for GIS," *ACM Sigmod Record*, vol. 23, no. 1, pp. 107–115, 1994.
[18] S. Aronoff, *Geographic Information Systems*. WDL Publications, Canada, 1989.
[19] G. Djuknic and R. Richton, "Geolocation and Assisted GPS," *IEEE Computer*, vol. 34, no. 2, pp. 123–125, 2001.
[20] A. Frank and M. Goodchild, "Two Perspectives on Geographical Data Modelling," Tech. Rep. 90-11, National Center for Geographic Information and Analysis, 1990.
[21] M. Goodchild, "Integrating GIS and Environmental Modeling at Global Scales," in *Proc GIS/LIS'91*, vol. 1, pp. 117–127, 1991.
[22] H. Couclelis, "People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS," in *Proc Intl. Conf. on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in Computer Science 639, pp. 65–77, 1992.
[23] J. Oliveira, *Projeto e Implementação de Interfaces para Sistemas de Aplicações Geográficas*. PhD thesis, IC-UNICAMP, december 1997. Orientação C. B. Medeiros.
[24] J. Oliveira and C. B. Medeiros, "A Software Architecture Framework for Geographic User Interfaces," in *Proceedings International Workshop on Emerging Technologies for Geo-Based Applications*, pp. 233–248, 2000.
[25] J. Oliveira and C. B. Medeiros, "User Interface Architectures, Languages and Models in Geographic Databases," in *Proc XI SBBD*, pp. 20–42, 1996.
[26] A. B. Prado, M. C. Baranauskas, and C. B. Medeiros, "Cartografia e Sistemas de Informação Geográfica como Sistemas Semióticos: Uma Análise Comparativa," in *Anais GEOInfo 2000 – II Workshop Brasileiro de GeoInformática*, pp. 88–95, 2000.
[27] M. Monmonier, *How to Lie with Maps*. University of Chicago Press, 1991.
[28] M. M. D. Calcinelli, "The Management of the Ambiguities in a Graphical Query Language for GIS," in *Proc. 2nd Symp. Spatial Database Systems*, pp. 99–118, Spring Verlag Lecture Notes in Computer Science 525, 1991.
[29] M. Egenhofer, "Spatial SQL: A Query and Presentation Language," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 1, pp. 86–95, 1994.
[30] N. Pissinou, K. Makki, and E. Park, "Towards the Design and Development of a New Architecture for Geographic Information Systems," in *Proc. 2nd Intl Conf. on Information and Knowledge Management- CIKM*, pp. 565–573, 1993.
[31] A. Voisard, "Designing and Integrating User Interfaces of Geographic Database Applications," in *Proc. ACM Workshop on Advanced Visual Interfaces*, pp. 133–142, 1994.
[32] P. Rigaux, *Interfaces Graphiques pour Bases de Données Spatiales: Application à la Représentation Multiple*. PhD thesis, CEDRIC - Conservatoire National des Arts et Metiers, 1995.
[33] Y. Kambayashi, T. Ishida, H. Tarumi, and K. Morishita, "Database with Space and Time Constraints for Digital Cities," in *Proc. International Workshop on Emerging Technologies for Geo-Based Applications*, pp. 297–307, 2000.
[34] R. Snodgrass, ed., *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995.
[35] S. P. Matias, "Processamento de Consultas ao Banco de Dados de Biodiversidade do BIOTA," Master's thesis, UNICAMP, december 2000. Orientação C. B. Medeiros.
[36] G. Camara, *Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos*. PhD thesis, INPE, dezembro 1995.
[37] M. Goodchild, R. Haining, and S. W. et al, "Integrating GIS and Spatial Data Analysis: Problems and Possibilities," *Intl. Journal of Geographical Information Systems*, vol. 6, no. 5, pp. 407–424, 1992.
[38] D. J. Maguire, M. F. Goodchild, and D. W. Rhind, eds., *Geographical Information Systems - volume I*. John Wiley and Sons, 1991.
[39] R. H. Guting, "An Introduction to Spatial Database Systems," *The VLDB Journal*, vol. 3, no. 4, pp. 357–400, 1994.
[40] D. J. Peuquet, "Toward the Definition and Use of Complex Spatial Relationships," in *Proc 3rd Intl. Symp. on Spatial Data Handling*, pp. 211–223, 1988.
[41] Z. Kemp and R. Thearle, " Modelling Relationships in Spatial Databases ," in *Proc 5th Intl. Symp. on Spatial Data Handling*, pp. 313–322, 1992. Volume 1.

[42] G.-C. Roman, "Formal Specification of Geographic Data Processing Requirements," *IEEE Transactions on Knowledge and Data Engineering*, vol. 2, no. 12, pp. 370–380, 1990.

[43] B. C. Ooi, *Efficient Query Processing in Geographic Information Systems.* Springer Verlag Lecture Notes in Computer Science, 1990.

[44] T. Joseph and A. Cardenas, "PICQUERY: A High Level Query Language for Pictorial Database Management," *IEEE Transactions on Software Engineering*, vol. 14, no. 5, pp. 630–638, 1988.

[45] G. Faria, C. B. Medeiros, and M. A. Nascimento, "An Extensible Framework for Spatio-Temporal Scientific Database Applications," in *Proc. Tenth Intl Conf. on Scientific and Statistical Database Management SSDBM*, pp. 202–206, 1998.

[46] G. Faria, "Um Banco de Dados Espaço-temporal para Desenvolvimento de Aplicações em Sistemas de Informação Geográfica," Master's thesis, UNICAMP, march 1998.

[47] C. Bonhomme, C. Trepied, M. Aufaure, and R. Laurini, "A Visual Language for Querying Spatio-Temporal Databases," in *Proc. 7th ACM Symposium on Advances in Geographic Information Systems*, pp. 34–40, 1999.

[48] D. Fernandes and A. Salgado, "GeoVisual Interface - a Visual Query Interface for Geographic Information Systems," in *Anais XV Simpósio Brasileiro de Bancos de Dados*, pp. 7–19, 2000.

[49] G. Graefe, "Query Evaluation Techniques for Large Databases," *ACM Computing Surveys*, vol. 25, no. 2, pp. 73–170, 1993.

[50] J. Orenstein, "Redundancy in Spatial Databases," in *Proc. ACM SIGMOD Conf.*, pp. 294–305, 1989.

[51] H.-P. Kriegel, " Query Processing in Spatial Database Systems ," in *Proc. 2nd Symp. Spatial Database Systems*, pp. 172–191, Springer Verlag Lecture Notes in Computer Science 525, 1991.

[52] G. Z. Silva, *Avaliação de Junções em Bancos de Dados Espaciais.* PhD thesis, COPPE-UFRJ, 1999.

[53] K. Chakrabarti and S. Mehrotra, "Efficient Concurrency Control in Multidimensional Access Methods," in *Proc ACM SIGMOD*, pp. 25–36, 1999.

[54] U. Dayal, M. Hsu, and R. Ladin, "A Transactional Model for Long-Running Activities," in *Proceedings 17th VLDB*, pp. 113–122, 1991.

[55] G. Alonso and A. El-Abbadi, " A Graph-based Model for Concurrency and Version Control in Geographic Information Systems Databases," Tech. Rep. 92-17, University of California - Santa Barbara, 1992.

[56] V. Kumar, A. Bugacov, M. Coutinho, and R. Neches, "Integrating Geographic Information Systems, Spatial Digital Libraries and Information Spaces for Conducting Humanitarian Assistance and Disaster Relief Operations in Urban Environments," in *Proc 7th ACM International GIS Symposium*, pp. 146–151, 1999.

[57] N. Paxton and O. Diaz, "Active Database Systems," *ACM Computing Surveys*, vol. 31, no. 1, pp. 63–103, 1999.

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

[59] H. Samet and W. Aref, "Spatial Data Models and Query Processing," in *Modern Database Systems. The Object Model, Interoperability and Beyond* (W. Kim, ed.), pp. 339–385, ACM Press, 1995.

[60] D. Lomet and B. Salzberg, "Access Methods for Multiversion Data," in *Proc ACM SIGMOD*, pp. 315–324, 1989.

[61] M. Nascimento and M. Dunham, "Indexing Valid Time Databases via B+ Trees," *IEEE TKDE*, vol. 11, no. 6, pp. 929–947, 1999.

[62] H. Gunadhi and A. Segev, "Efficient Indexing Methods for Temporal Relations," *IEEE TKDE*, vol. 5, no. 3, pp. 496–509, 1993.

[63] E. Hoel and H. Samet, "A Qualitative Comparison Study of Data Structures for Large Line Segment Databases," in *Proc ACM SIGMOD Conf.*, pp. 205–214, 1992.

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

[65] D. Rotem, "Spatial Join Indices," in *Proc. IEEE Data Engineering Conf.*, pp. 500–509, 1991.

[66] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith, "The Sequoia 2000 Benchmark," in *Proc ACM SIGMOD Conf.*, pp. 2–11, 1993.

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

[68] A. Carneiro and G. Magalhaes, "Usando Dados Geográficos Urbanos na Comparação de Desempenho de Métodos de Acesso Espaciais," in *Anais XIII Simpósio Brasileiro de Bancos de Dados*, pp. 401–416, 1998.

[69] B. Salzberg and V. Tsotras, "Comparison of access methods for time-evolving data," *ACM Computing Surveys*, vol. 2, no. 31, pp. 158–221, 1999.

[70] J. O. Silva, "Geração e indexação de dados espaço-temporais," Master's thesis, IC - UNICAMP, 1999. Orientação M. Nascimento.

[71] G. Proietti and C. Faloutsos, "Analysis of Range Queries and Self-spatial Join Queries on Real Region Datasets Stored Using an R-Tree," *IEEE TKDE*, vol. 12, no. 5, pp. 751–762, 2000.

[72] G. Z. ao, J. Souza, R. Monteiro, and L. Azevedo, "A Raster Approximation for Processing of Polyline Joins," in *Anais XV Simpósio Brasileiro de Bancos de Dados*, pp. 130–142, 2000.

[73] G. Gaede and O. Gunther, "Multidimensional Access Methods," *ACM Computing Surveys*, vol. 30, no. 2, pp. 170–271, 1998.

[74] N. Gervaerdt, " Avaliacao de Desempenho de Estruturas de Acesso a Dados Hiper-dimensionais," Master's thesis, UNICAMP, 2000. Orientação M. Nascimento.

[75] L. del Val, N. Leite, and C. Medeiros, "An Architecture for Content-based Retrieval of Remote Sensing Images," in *IEEE International Conference on Multimedia and Expo (ICME-2000)*, july 2000.

[76] R. SantosFilho, A. Traina, C. Traina, and C. Faloutsos, "Similarity Search Without Tears: the Omni Family of All Purpose Access Methods," in *Proc. ICDE Conference*, 2001.

[77] X. Xu, J. Han, and W. Lu, "RT-Tree: an Improved R-tree Index Structure for Spatiotemporal Databases," in *Proc. 4th Symposium on Spatial Data Handling - SSD*, 1990.

[78] G. Jomier, M. Manouvrier, and M. RUkoz, "Storage and Management of Similar Images," *Journal of the Brazilian Computer Society*, vol. 3, no. 6, pp. 13–25, 2000.

[79] M. N. Y. Theodoridis, "Generating Spatiotemporal Datasets on the WWW," *ACM Sigmod Record*, vol. 29, no. 3, pp. 39–43, 2000.

[80] M. Nascimento, J. Silva, and Y. Theodoridis, "Evaluation of Access Structures for Discretely Moving Points," in *Proc. Workshop Spatio-Temporal Database Management*, pp. 171–188, 1999.

[81] S. Saltenis, C. Jensen, S. Leutenegger, and M. Lopez, "Indexing the Positions of Continuously Moving Objects," in *Proc. ACM SIGMOD Conference*, pp. 331–342, 2000.

[82] D. Abel, "A Model for Data Set Management in Large Spatial Information Systems," *Intl. Journal of Geographical Information Systems*, vol. 3, no. 4, pp. 291–302, 1989.

[83] R. Newell, D. Theriault, and M. Easterfieldy, "Temporal GIS - modeling the evolution of spatial data in time," *Computers and Geosciences: An Intl. journal*, vol. 18, no. 4, pp. 427–434, 1992.

[84] M. P. Worboys, "A Model for Spatio-temporal Information," in *Proc 5th Intl. Symp. on Spatial Data Handling*, pp. 602–611, 1992. Volume 2.

[85] M. Batty and Y. Xie, "Modelling inside GIS: Part I. Model Structures, Exploratory Spatial Data Analysis and Aggregation," *Intl. Journal of Geographical Information Systems*, vol. 8, no. 3, pp. 291–308, 1994.

[86] M. Worboys, "A Unified Model for Spatial and Temporal Information ," *The Computer Journal*, vol. 37, no. 1, pp. 26–33, 1994.

[87] F. L. Pimentel, "Uma Proposta de Modelagem Conceitual para Dados Geográficos – o Modelo MGeo+," Master's thesis, UFPE, october 1995.

[88] M. Erwin, R. Guting, M. Schneider, and M. Vazirgiannis, "Spatio-temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases," Tech. Rep. 224-12/97, FernUniversitat Hagen, 1997.

[89] C. Parent, S. Spaccapietra, and E. Zimanyi, "Spatio-Temporal Conceptual Models: Data Structures + Space + Time ," in *Proc. 7th ACM Symposium on Advances in Geographic Information Systems*, pp. 26–37, 1999.

[90] D. Pfoser, *Issues in the Management of Moving Point Objects*. PhD thesis, University of Aalborg, 2000. Supervisor C. Jensen.

[91] S. Papastavrou, G. Samaras, and E. Pitoura, "Mobile Agents for World Wide Web Distributed Database Access," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 5, pp. 802–820, 2000.

[92] L. Forlizzi, R. Guting, E. Nardelli, and M. Schneider, "A Data Model and Data Structures for Moving Objects Databases," in *Proc. ACM SIGMOD Conference*, pp. 319–330, 2000.

[93] V. Mesev, "Remote Sensing of Urban Systems: Hierarchical Integration with GIS," *Computers, Environment and Urban Systems*, 1997.

[94] Interop, "International Conference and Workshop on Interoperating Geographic Information Systems." Web site http://www.ncgia.ucsb.edu/conf/interop97, 12 1997. Site address valid as of 07/99.

[95] P. Yates and I. Bishop, "The Integration of Existing GIS and Modelling Systems: with Urban Applications," *Computers, Environment and Urban Systems*, vol. 22, no. 1, pp. 71–80, 1998.

[96] S. Keller, " Modeling and sharing geographic data with INTERLIS," *Computers and Geosciences*, vol. 25, no. 1, pp. 49–59, 1999.

[97] F. Wang and S. Jusoh, "Integrating Multiple Web-based Geographic Information Systems," *IEEE Multimedia*, pp. 49–61, jan-mar 1999.

[98] T. Barclay, J. Gray, and D. Slutz, "Microsoft TerraServer: a Spatial Data Warehouse," in *ACM SIGMOD Conference*, pp. 307–318, 2000.

[99] S. Lardon, T. Libourel, and J. Cheylan, "Concevoir la Dynamique des Entités Spatio-temporelles," *Revue Internationale de Géomatique*, vol. 9, no. 1, pp. 45–66, 1999.

[100] M. Ranzinger and G. Gleixner, "GIS Datasets for 3D Urban Planning," *Computers, Environment and Urban Systems*, vol. 21, no. 2, pp. 159–173, 1997.

[101] H. J. G. L. Aalders, "Quality Metrics for GIS," in *Advances in Gis Research II* (M. J. Kraak and M. Molenar, eds.), pp. 5B1–5B10, 7th International Symposium on Spatial Data Handling, Delft University, August 1996.

[102] W. Michener, J. Brunt, J. Helly, T.Kirchner, and S. Stafford, "Nongeospatial Metadata for the Ecological Sciences," *Ecological Applications*, vol. 7, no. 1, pp. 330–342, 1997.

[103] S. C. Guptill, "Building a Geospatial Data Framework - Finding the Best Available Data," in *Data Quality in Geographic Information - From Error to Uncertainty* (M. Goodchild and R. JeanSoulin, eds.), ch. Quality Concepts and Models, pp. 31–36, Hermes, 1998.

[104] F. G. D. Committee, "Content Standards for Digital Geospatial Metadata." http://geochange.er.usgs.gov/pub/tools-/metadata/standard/metadata.html, 1994.

[105] F. Harvey, "Quality Needs More than Standards," in *Data Quality in Geographic Information - From Error to Uncertainty* (M. Goodchild and R. JeanSoulin, eds.), ch. Quality Concepts and Models, pp. 37–42, Hermes, 1998.

[106] C. B. Medeiros and A. Alencar, "Qualidade de Dados e Interoperabilidade em SIG," in *Anais I GEOINFO – I Workshop Brasileiro de GeoInformática*, pp. 12–19, 1999.