

Inconsistency Issues in Spatial Databases

Andrea Rodríguez

Department of Computer Science
University of Concepción, Chile
andrea@udec.cl
<http://inf.udec.cl/~andrea>

Abstract. This chapter analyzes inconsistency issues in spatial databases. In particular, it reviews types of inconsistency, specification of integrity constraints, and treatment of inconsistency in multiple representations and data integration. The chapter focuses on inconsistency associated with the geometric representation of objects, spatial relations between objects, and composite objects by aggregation. The main contribution of this paper is a survey of existing approaches to dealing with inconsistency issues in spatial databases that emphasizes the current state of the art and that outlines research issues in the context of inconsistency tolerance.

1 Introduction

During the past several years traditional databases have been enhanced to include spatially referenced data. This type of data is an essential component of existing applications such as Geographic Information Systems (GIS), Computer-Aided Design (CAD), multimedia information systems, data warehousing, and NASA's Earth Observing System (EOS).

Spatial databases have been defined as database systems with a model and query language that support spatial data types and provide spatial indexing and efficient algorithms for spatial query processing [37]. Unlike classical database theory, where the content of databases is abstract, in spatial databases the content has some interpretation and laws of real geometry hold. This interpretation induces to much diverse classes of data structures and data manipulations. Spatial databases have no clear separation between what is handled by the database management system (DBMS) and what is handled by the software application [48]. For example, it is not always clear whether or not an operation that finds the shortest path in a network is part of the spatial DBMS. Consequently, there is no consensus of what properties and features should be part of spatial data manipulation languages.

In spatial databases, theory about spatial information is used to define *spatial data models* or *geomatic models*. Spatial data models represent information about the n -dimensional real space \mathbb{R}^n , a space that is infinite and cannot be represented with an extensional data model. Operations in spatial databases may or may not depend on the spatial data model underlying the data representation, an issue related to the concept of *genericity* that was introduced in

classical databases [13] and then applied in the domain of spatial databases by Paredaens *et. al* [47] [48].

The integration of spatial data into traditional database systems requires addressing nontrivial issues at various levels. They range from ontological issues about the conceptualization of space to more technical issues about access mechanisms and file management [58]. As consequence, progress in SDBMS is the result of an interdisciplinary research effort. The treatment of inconsistency of spatial data also requires an interdisciplinary approach. Consistency of spatial information must deal with ontological issues concerning physical reality [18] [34] (e.g., an object can only have one physical location at a time). It needs to consider the appropriate conceptual frameworks for analyzing spatial consistency [24] [30] [43] [63], such as models for consistency at multiple representational levels or granularities. It also concerns the specification language of integrity constraints [9] [41] and the design of computational-geometry algorithms to implement consistency checkers.

This chapter analyzes inconsistency issues in spatial databases. Its main contribution is a survey of existing approaches to dealing with inconsistency issues that emphasizes the state of the art and outlines research issues in the context of inconsistency tolerance. The chapter focuses on the *geometric representation* of objects (i.e., location and shape), *spatial relations* between objects, and *composite objects by aggregations*. Positional information is often imprecise in spatial databases, which may result in conflicting geometric representations of objects (i.e., two different geometric representations for the same object). Spatial relations play an important role in spatial databases, since they are usually the basis for specifying integrity and query constraints [28]. *Spatial relations* are typically derived from positional information; however, they may not be affected by conflicting objects' geometric representations because objects may hold the same spatial relation in these representations [53]. *Nested aggregations* are fundamental abstraction mechanisms for modeling spatial phenomena. For example, countries contain states that contain counties. Aggregations impose requirements for data modeling and data consistency with respect to the relationships between parts and wholes.

The organization of this chapter is as follows. Section 2 gives a brief overview of spatial databases. Section 3 discusses the types and sources of inconsistency in spatial information. Section 4 discusses the specification of constraints, consistency at multiple representational levels, and consistency in the integration of spatial information. Section 5 addresses inconsistency tolerance of spatial information. Final conclusions are given in Section 6.

2 Spatial Database Overview

Spatial database systems consist of data about objects and properties in the world with respect to their locations [54]. These systems deal with diverse kinds of data, from natural to man-made features, which demands specific models that both capture the semantics of spatial data and also offer a high level of

abstraction. At an abstract level, spatial objects can be *atomic* or *complex*. *Atomic spatial objects* are composed of a description and a spatial-component (e.g., a landparcel has a code number and a geometric component represented by a surface), and by aggregation, *complex spatial objects* are composed of a description and a set of spatial objects (e.g., a sport club may be composed of a sport field, tennis court, gymnasium, and so on).

Abstractions that need to be supported in a SDBMS are *partitions* and *networks* [37]. A partition represents either a spatial feature or space cell. Partitions are commonly used to represent thematic layers or maps (e.g., soil-type maps and administrative boundaries). A network is seen as a graph embedded in the plane with nodes (e.g., places) that are connected by edges (e.g., highways, rivers, channels, and so on). Other collections of spatial objects that are often relevant to spatial databases are *nested partitions* (e.g., a country is an aggregation of states and a state is an aggregation of counties) and *triangular irregular networks* (TIN) (e.g., terrain digital models).

Applications of spatial information, in particular geospatial applications, differ from traditional data applications for the following reasons [65]:

- Spatial information deals with spatial and non-spatial data, where the definition of spatial data types should be closed under the operations applicable to them.
- Data are highly structured by the notion of object aggregation.
- The existence of user-defined operations that require an extensible underlying model.
- Functions exist at both a low-level of abstraction (e.g., functions over points, lines and polygons) and a high-level of abstraction (e.g., functions over maps and configurations).

As an example of what is the kind of data that are modeled in a spatial database, consider the case of a land information system (LIS) composed of *landparcels* and information related to *land ownership* (Figure 1). A landparcel is a spatial object that has a spatial component (e.g., a landparcel may be represented by a surface) and attributes describing properties of the land (e.g., identification, owner, use, and so on). Aggregations of landparcels are *sections* in the LIS. Geometric operations may be defined at the level of an individual landparcel (e.g., the area of a landparcel) or at the level of thematic maps (e.g., the merge of landparels with topographic information). In addition to operations handled by the data manipulation language, a user may need to define a new operation over the landparcels' geometries. For example, a user may want to define a function that detects particular shapes of landparcels. This implies that the underlying data model must allow the definition of new types of operations.

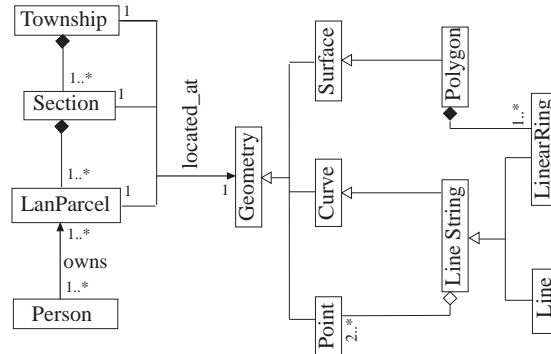


Fig. 1. A conceptual model of a portion of a land information system in UML

2.1 Spatial Data Models or Geomatic Models

Spatial modelers often make the classical distinction between *field-based* and *entity-based* view of the space [22] [59]. In the field-based view of the space, each point in the space has one or more attribute values that are typically defined by continuous functions in coordinates x and y (e.g., temperature, altitude, and pollution). The view of the space is a continuous *field* that represents a phenomenon whose attribute values vary with the position in the space. In this view, the concept of entity or object is irrelevant. In the entity-based view of the space, by contrast, space is composed of spatial objects that are entities with explicit identity. Each of these views of space can be represented by using different spatial data models.

Spatial data models depend on the operations that have to be defined and the efficiency needs of the implementation. One of the simplest and common models is the *spaguetti model* or *vector model* [54]. Although the *vector model* is usually associated with an entity-based view of the space (e.g., Figure 1), it can also model a field-view of the space (e.g., a digital elevation model that is represented by a triangulated irregular network TIN). This model has efficient algorithms for detecting properties of spatial objects (e.g., overlapping, intersection, and spatial inclusion). In this model, the information in a n -dimensional space is represented by using m -dimensional geometric primitives, with $m < n$. The common types of primitives used in this model are, where $\langle \rangle$ are lists, $[]$ are tuples and $\{\}$ are sets:

- *Points or zero-dimensional primitives* (e.g., the locations of utility poles can be represented by points): $[x : real, y : real]$.
- *Polylines or one-dimensional primitives*, whose data structure is a finite list of points (e.g., the access roads to landparcels can be represented by one-dimensional primitives) : $\langle point \rangle$.

- *Polygons or two-dimensional primitives* are also represented by a list of points, but this list represents a non self-intersecting closed polylines (Figure 2) (e.g., the spatial component of a landparcel is described by a two-dimensional primitive): $\langle \text{points} \rangle$. By aggregation, *complex polygons or regions* are sets of polygons: $\{\text{polygon}\}$.

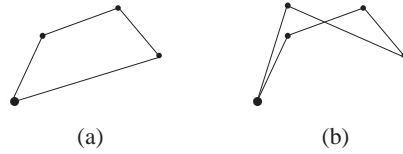


Fig. 2. Polylines: (a) closed and non self-intersecting polyline and (b) closed and self-intersecting polyline

Using the Vector Model, in a two-dimensional space, for example, any spatial object is presented by points or polylines, which are considered zero- and one-dimensional geometric primitives, respectively (Figure 3). In a three-dimensional space, a polyhedra is represented by the boundaries of which contain planar facets (i.e., surfaces), polylines, and points.

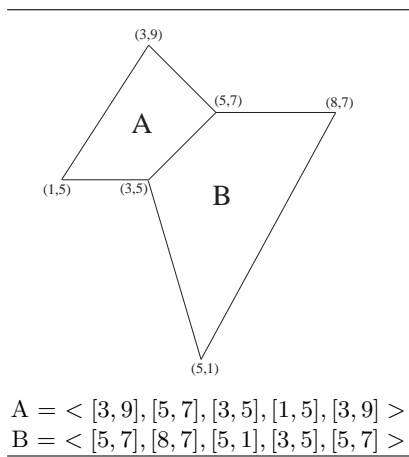


Fig. 3. An example of the Vector Model

Other types of models that concern with practical issues of efficiency are the *raster model* and the *piano model* [38] [39] [55], which are often, but not always, seen as the typical way to model a field view of the space. The *raster model* intentionally represents spatial information by a finite number of cells or raster points, where the infinite number of points associated with a cell share the same properties. The main problem of this model is the needed approximation of geometric elements to raster points of cells (Figure 4). The *piano model* combines techniques of space-filling curves and quadtrees [55]. This model encodes a linear order of cells that partition a space while maintaining locality (i.e., cells close to each other in the space are also close to each other in the linear order). This linear order is done recursively for a grid that is obtained by hierarchical subdivision of the space (Figure 4).

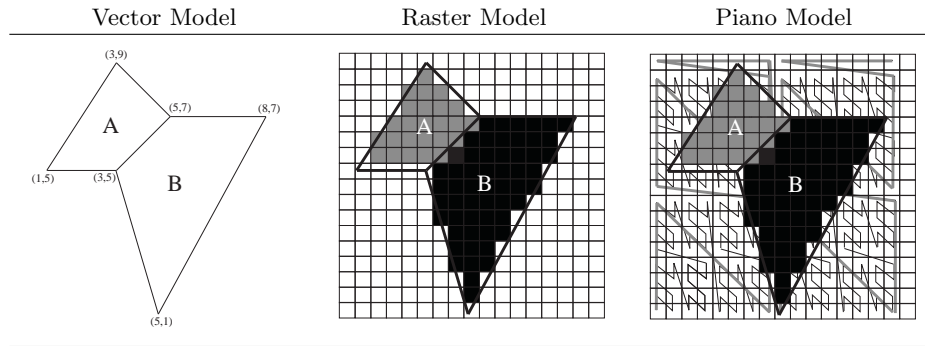


Fig. 4. The Vector Model, Raster Model, and Piano Model

Focusing on theoretical issues of an entity-based view of the space, other models are the *topological model* and the *constraint model*. The *topological model* addresses data manipulation that is topological in nature. This type of data manipulation involves concepts such as adjacency, connectivity and containment. For example, a query that can efficiently solve in this type of models is “find landparcels that are adjacent to the landparcel whose identifier is equal to X .”

A topological model can be seen as a planar network, with the following primitives of interest:

- *Points* are pairs of real numbers: $[x : real, y : real]$.
- *Nodes* are tuples composed of a point and a list of arcs in which the node is one of the extremes: $[point, < arc >]$.
- *Arcs* are tuples composed of a starting node, ending node, left polygon, right polygon and list of internal points of the arcs : $[start_node, end_node, left_polygon, right_polygon, < point >]$.
- *Polygons* are lists of arcs $< arc >$.

– *Regions* are sets of polygons $\{polygon\}$.

To make clear the difference between the Vector model and Topological model, consider the same spatial objects represented with these two models in Figure 5. The difference between the two models is that the topological model handles explicitly common boundaries and adjacency between polygons.

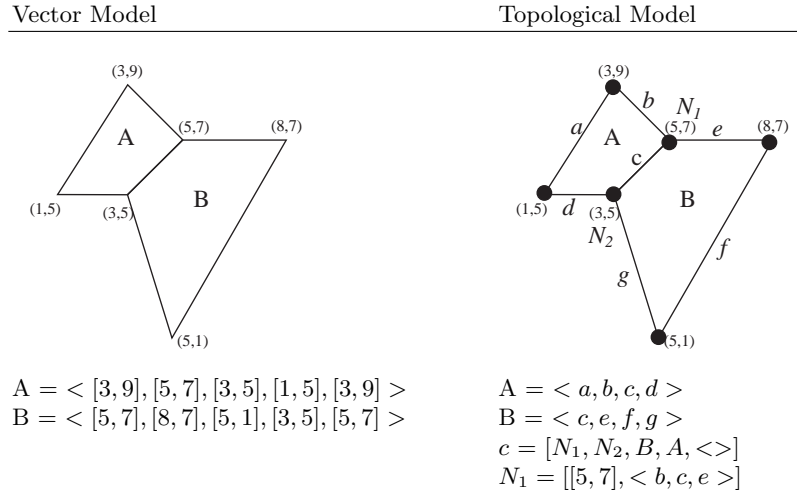


Fig. 5. Comparing representations of two landparcels in the Vector Model and Topological Model

The *constraint model* defines any geometrical figure by an elementary geometry expressed by first-order logic over the real numbers [42]. The constraint data model aims to handle infinite relations (i.e., infinite sets of points in a space), which are represented by quantifier-free formulas. For example, consider the same objects *A* and *B* in Figure 5, the corresponding representations in the constraint model are:

Object	Constraint-based Representation
<i>A</i>	$y \geq 5 \wedge y \leq 2x + 3 \wedge y \leq -x + 12 \wedge y \leq x + 2$
<i>B</i>	$y \leq 7 \wedge y \leq x + 2 \wedge y \leq 2x - 9 \wedge y \leq -2x + 11$

In addition to the geometric representation of spatial objects (i.e., position and shape of objects), spatial relations between objects play an important role in spatial information systems, since such relations refer to the way people perceive, reason, and describe spatial information in a variety of languages [28]. Models of spatial information may be more or less efficient to determine spatial relations. Positional information is often used for determining the spatial relations

between objects and, therefore, these relations can be determined when spatial data models, such as the vector or raster models, are used. Spatial relations such as adjacency and containment, however, do not require absolute positional data [11] and are efficiently handled with the topological model. For example, one could say that two objects meet because they share a common boundary, disregarding the exact location of the objects.

Common spatial relations are typically grouped into three kinds: *topological*, *orientation*, and *distance* [61] [66]. *Topological relations* deal principally with the concept of connectivity and are invariant under topological transformations, such as rotation, translation, scaling. *Orientation relations* presuppose the existence of a vector space and are subject to changes under rotation, while they are invariant under translation and scaling. *Distance relations* express spatial properties that reflect the concept of a metric and, therefore, they change under scaling, but are invariant under translation and rotation. Among these spatial relations, topological relations have spurred much recent research [17] [20] [25] [28] [46]. They are considered to capture the essence of a spatial configuration – topology matters, metric refines [11].

In summary, spatial databases deal with objects that have a position in a space as well as with spatial relations among these objects. Different models of spatial information exist that address the geometric representation of spatial objects, some of them concerning theoretical issues and others concerning issues of efficiency. It is still a research challenge to create models for spatial information that combine a solid theoretical foundation with efficiency considerations.

2.2 Data Model and Query Language

The previous Section has reviewed models for the geometric representation of spatial objects. Such models have been integrated into traditional database management systems to profit from well established data models and data structures of traditional database systems. This Section concentrates on the extended relation model, one of the possible data models that supports the representation and querying of spatial objects. The extended relational model is the widest used model in current spatial database management systems. Descriptions of other models, such as the object-oriented data model and the constraint data model can be found in [42][44][59][65].

In extended relational systems, end users manipulate values whose types are basic, such as integer or characters, but also abstract data types (ADT) that are accessible through the operations defined on them [35] [62]. In these systems, each type of spatial objects corresponds to a relation that contains a geometric-type attribute, such as *region* or *line*, among others. A link between relations is handled through the standard mechanism of relational schemas; i.e., by means of a foreign key.

Consider, for example, the cadastral application system presented in Figure 1. The corresponding data schema in the SQL data definition language (DDL) is:


```

create table Township(town_code: integer, name: string
                      geometry: region, Primary Key (town_code))
create table Section(town_code: integer, section_code: integer,
                      geometry: region, Primary Key(section_code),
                      Secondary Key(town_code))
create table LandParcel(section_code: integer, parcel_code: integer,
                          geometry: region, Primary Key(parcel_code),
                          Secondary Key(section_code))
create table Person(person_id: integer, name: string,
                     Primary Key(person_id))
create table Ownership(person_id: integer, parcel_code: integer,
                        Primary Key(person_id,parcel_code),
                        Foreign Key(person_id), Foreign Key(parcel_code))

```

A spatial selection query in SQL based on the previous schema could be “find the identifier of the town that contains the landparcel whose parcel_code is equal to X ”:

```

select  t.town_code from Township t, LandParcel l
          where l.parcel_code= 'X' and inside(l.geometry,t.geometry)

```

The answer to this query will be inconsistent if two or more towns’ identifiers (*town_code*) are retrieved, since a landparcel must only be part of one town. A more complex query is, for example, “create a map and retrieve the area from the aggregation of landparcels grouped by sections.” Such query could be expressed in SQL as:

```

select   area(o.geometry), sum(l.geometry) from LandParcel l, Section o
          where o.section_code = l.section_code group by l.section_code

```

The query answer is inconsistent if the area of the aggregation is different to the area of the spatial component of section (i.e., if $area(sum(l.geometry)) \neq area(o.geometry)$), since the aggregation of the geometric parts should be equal to the geometric whole.

3 Types and Sources of Inconsistency of Spatial Information

Spatial information systems often must deal with different kinds of data imperfections, which can be classified into *uncertainty*, *imprecision/vagueness*, *incompleteness*, and *inconsistency* [7] [10] [49]. Uncertainty is a kind of data imperfection that arises from the lack of information about the state of the world (e.g., “if the distance between Santiago and Concepción is unknown, the time that takes to travel from Santiago to Concepción is uncertain”); imprecision is a kind of data imperfection that arises from the granularity of the language used to make

an imprecise statement (e.g., “Santiago is located in America”); vagueness is a kind of imprecision that arises from the use of terms when there are cases for which it is difficult to decide if they are covered or not by a particular concept (e.g., “Santiago is *close* to Concepción”); incompleteness is a kind of imperfection that arises from the absence of some data values (e.g., a missing road in a transportation network); and inconsistency is a kind of data imperfection that arises from the coexistence of two contradictory facts (e.g., “Concepción is located at 500 km from Santiago” and “Concepción is located at 600 km from Santiago”).

From an ontological perspective, Frank [34] distinguishes consistency rules that capture the meaning of space and time. At a bottom level, the physical reality, which is independent of human-perception, satisfies “natural laws,” rules that are thought to be universal; for example, the speed of an object is related to the acceleration. At the physical-observation level; that is, the physical reality observed through instruments, data should follow the distribution of measurement values according to the expected error. For example, the distance that is measured by an instrument must not be too different from the calculated distance between two stored points. At the object-property level, objects should satisfy necessary conditions. For example, a stadium must be composed of a sport field. At the social-definition level, context constrains the consistency of data in the form of X counts as Y in context Z . For example, a historical building is a building older than 150 years, but this is true in the context of Chile. Finally, at the cognitive-agent level (i.e., agents are people or organizations), there should be no contradiction with respect to the common understanding of reality by an agent. For example, an organization (i.e., an agent) is composed of sub-agents that are departments of the organization. Consistency rules at this level enforce that each of the sub-agents behaves consistently with the organization’s view of the world.

Considering ideas from [18] [69], spatial inconsistencies can be related to, but they are not the same than, forms of error. From the perspective of the type of characteristics the inconsistency refers to, inconsistency is related to what are called *primary* or *secondary* forms of error. The primary form of error corresponds to a wrong description of location or characteristics/qualities of spatial objects. A typical case is the conflicting geometric representation of a spatial object; for example, having an integrity constraint that states that objects have only one location, there is an inconsistency derived from a primary type of error if there exist more than one location for a spatial object. This type of inconsistency occurs because there exist differences in data accuracy or precision, but also because many observations of spatial phenomena are essentially vague. For example, the boundaries of cities, mountains, and oceans cannot be determined with precision, which may make two observers record two different locations for the same object.

In general, an inconsistency associated with a primary error violates a basic principle of *location or attribute uniqueness*. In spite of the desirable condition of positional uniqueness, spatial information often deal with inaccurate coordinates

and imprecise data. Topological facts; however, may not require data about positions of objects [53] to be consistent. There may exist different geometric representations of objects (i.e., inconsistency); however, the spatial relations between objects may be the same in these representations (Figure 6).

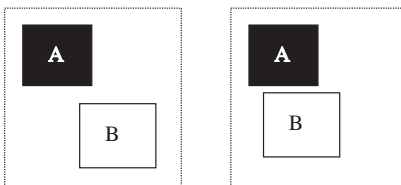


Fig. 6. A configuration of two objects with two different geometric representations, but with the same topological relation *disjoint* between objects

A spatial inconsistency related to a secondary error refers to a contradiction between stored data and constraints associated with *structural* definitions of geometric primitives. For example, a surface must be bounded by closed and non self-intersecting polylines. Inconsistency may also be related to *semantic* contradictions, such as when a road overlaps a body of water. These types of inconsistency, structural or semantic, depend on the spatial domain, and they are captured by rules that should be expressed within the data model.

Some relevant characteristics of spatial applications that should be considered in the treatment of consistency are [8] [53]:

- Spatial information deals with spatial and non-spatial data. In addition to inconsistency of non-spatial data, inconsistency may occur between spatial and non-spatial or within spatial data.
- Many spatial data are inherently vague, which may lead to conflicting data. Vagueness may make observations of a same spatial phenomenon be different and, therefore, have conflicting representations.
- Topological and other spatial relations are very important and are usually implicitly represented. Spatial relations are typically derived through data manipulation such that checking topological inconsistency involves not only to check stored facts in a database; but also to check for results of data manipulation.
- A modification in a spatial database may cause simultaneous updates in a large number of records. Depending on a spatial representation, a modification of an object's boundary may affect the representation of its neighboring objects' boundaries as well. For example, two partitions of the space that share a common boundary may need an update at the same time when one of them changes its boundaries; otherwise, partitions could overlap, which contradicts the definition of partitions of a space.

- Spatial databases may need to treat different levels of detail in the spatial representation. These representations may be handled as duplicate information or may be generated dynamically through a generalization process. For example, you may need to keep the representation of a city as a region and a point, depending on the visualization needs of an application. Since duplication of information may occur, it is necessary to keep consistency of multiple representational levels (e.g., a region cannot be a line at a coarse representation).
- Many queries are defined in terms of combinations of functions that exist at both a low-level of abstraction (e.g., geometry types) and a high-level of abstraction (e.g., maps, configurations). For example, a query may be to select the location of a lanparcel or may be to obtain a map by the merge of lanparcels with transportation networks.

As a conclusion, differences between traditional databases and spatial databases are based on the interpretation of data. The spatial domain brings up different types of inconsistency that may require ad-hoc treatments. A contradiction of facts in a traditional database is commonly determined by the property of equality of attribute values. In spatial databases, however, a spatial attribute (e.g., a region that represents a spatial object) is not only a single value, it underlies a model of the space composed of a number of geometric primitives. In this context, data consistency does not only concern with the comparison of spatial attribute values, but also, the analysis of contradictions between the stored data and the model of spatial information (e.g., a polygon that is represented by a self-intersecting polyline contradicts the classical model where a polygon is defined by a closed and non self-intersecting polyline).

4 Work on Consistency in Spatial Databases

Research in the area of consistency in spatial databases has tried to clarify concepts about types of consistency, incorporate integrity constraints at different levels of the database design, and conceptualize consistency problems in generalization and information-integration processes. In all cases, the research effort has focused on how to detect or prevent inconsistencies. Although issues about inconsistency tolerance have been addressed for traditional relational databases, spatial databases have not handled explicitly inconsistency tolerance in query answering.

4.1 Integrity constraints in Spatial Databases

Inconsistency arises when integrity constraints are violated. Thus, constraints must be taken into account when updating a database so that the semantics and quality of data are preserved. In the spatial domain, integrity constraints have been mainly used for preventing structural inconsistency (i.e., inconsistency

between stored data and rules of geometric primitives), whereas conflicting information about positional information has been treated as a problem of data accuracy.








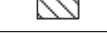
In addition to traditional integrity constraints concerning static, transition, and transactional aspects of databases systems [31], rules about spatial data must ensure consistent updating of spatial information (i.e., consistency of the geometric representation of objects with respect to a model of spatial information). A typical classification of these spatial constraints is [18] :

- *Topological constraints.* Topological constraints are those constraints that address geometrical properties and spatial relations. They may be associated with structural considerations, such as that partitions only meet or are disjoint, or topological conditions, such as centerlines must meet at intersections. Considering a subset of topological constraints, Servigne *et al.* [57] defined *topo-semantic* constraints as those that relate geometry with semantic conditions, as in the constraint that a city’s administrative region must be contained within its corresponding city limits.
- *Semantic integrity constraints.* These constraints are concerned with the meaning of geographic features; for example, landparcels are not contained in building_blocks.
- *user-defined integrity constraints.* These types of constraints are equivalent to business rules in non-spatial DBMS; for example, legal rules that constraints the installation of a gas station in a given region.

Like in traditional database systems, constraints at a conceptual and logical level in spatial databases are inherited by the implementation or physical level. These constraints are translated into a proprietary scripting language or into explicit constraints coded in the application programs [31]. At a logical level, Hadzilacos and Tryfona [41] describe a logical model with definitions of constraints based on topological relations. They state that it is possible but cumbersome to define topological constraints based on absolute positions. Therefore, they use a formal framework for defining topological relations [25] [27] upon which integrity constraints are specified. This framework defines topological relations between subsets of a classical topological space by the emptiness or non-emptiness of the two-by-two intersections of the subsets’ interiors (\circ) and boundaries (δ). Table 1 summarizes the resulting eight possible topological relations between two polygons. This table indicates, for example, that a *disjoint* relation exists when the intersections between boundaries, between interiors, between boundary and interior, and between interior and boundary are the empty set.

Within Hadzilacos and Tryfona’s framework [41], spatial relations and integrity constraints are expressed by using first-order logic. Atomic topological formulae in combination create topological sentences. Atomic topological formulae include geometric operators over objects, elementary topological relations between objects, and comparison between objects’ attributes. For example, consider the following statement in natural language of a semantic integrity constraint in a cadastral application: *land-parcels are not contained in building_blocks*. The formal specification of this constraint for land_parcels lp and

Table 1. Definition of topological relations between regions

	$\delta\delta$	$\circ\circ$	$\delta\circ$	$\circ\delta$	Relation
<i>disjoint</i>	\emptyset	\emptyset	\emptyset	\emptyset	
<i>meet</i>	$\neg\emptyset$	\emptyset	\emptyset	\emptyset	
<i>overlap</i>	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	
<i>cover</i>	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	\emptyset	
<i>covered_by</i>	$\neg\emptyset$	$\neg\emptyset$	\emptyset	$\neg\emptyset$	
<i>contain</i>	\emptyset	$\neg\emptyset$	$\neg\emptyset$	\emptyset	
<i>inside</i>	\emptyset	$\neg\emptyset$	\emptyset	$\neg\emptyset$	
<i>equal</i>	$\neg\emptyset$	$\neg\emptyset$	\emptyset	\emptyset	

building_blocks *bl* based on the topological relations defined in Table 1 is:

$$\forall(lp, bl) [\neg inside(lp, bl) \wedge \neg covered_by(lp, bl)] \quad (1)$$

Some topological constraints define geometric primitives or some spatial dependences of composite objects. Consider, for example, partitions of a space. To define a partition rule in first-order logic, one needs to consider predicates of the type $P_i(x)$, with x being an interior point of an object P_i . The spatial aggregation of partitions $P_0() \dots P_n()$ into $W()$, assuming that partitions can only *meet* or be *disjoint*, where *meet* and *disjoint* were defined in Table 1:

$$\forall(P_i, P_j) [meet(P_i, P_j) \vee disjoint(P_i, P_j)] \quad (2)$$

is then defined by the statement that a point x in the aggregation must belong to one partition $P_i()$:

$$\forall(x) [W(x) \equiv (P_0(x) \vee P_1(x) \vee \dots \vee P_n(x))] \quad (3)$$

A graph-based model of maps has also been used to establish topological integrity constraints of objects and their aggregations as a map [53]. This model makes it possible to guarantee the consistency of a map through database updates with respect to a set of topological constraints over *vertices*, *edges* and *faces* on the map graph. These integrity constraints are equivalent to the mathematical axioms of maps that are defined by a graph that is plane, connected, nonseparable and formed by edges that are straight lines bounding internal faces.

Some attempts have been made to provide end users with easy mechanisms that hide the logic in specifying constraints [19] [52] [57]. An early work by

Pizarro *et al.* [52] presents a visual language that depicts unacceptable database states. This visual language can then generate first-order predicates of spatial constraints. Another study allows users to define constraints in an English-like fashion. Basic components of the language are entity classes, relations, and qualifiers (e.g., forbidden, at least n times, at least most n times, or exactly n times) [57]. Following the same idea, Crockcroft's work [19] extends the previous specification to include attribute values in the topological constraints. For example, a butterfly valve must not intersect a pipe if the diameter of the pipe is greater than 40 inches. This interface for end-users is a standalone software tool that is integrated with a Geographic Information System (GIS).

4.2 Consistency at Multiple Representational Levels

The problem of multiple representations consists of data changing their geometric and topological structure due to changes in scale. Conceptually, multiple representations may be considered as different data sets that cover the same area with different levels of detail. Within the context of assessing consistency at multiple representations, topological relations are considered to be first-class information, which must prevail in case of conflicts [24] [29] [30] [43]. This means that, at different scales, there is no inconsistency in having different geometric representations of a same object if some topological constraints are satisfied.

Initially, topological consistency was treated at the low level of data structures, counting nodes and arcs to assure that an object's topology is complete [45]. This strategy accounts for changes in the geometry of objects, but it does not assure consistency of the relations between objects. For example, it does not handle consistency of the topological changes that may occur when, at a coarse representation, several parts become a single object or when holes of objects disappear. Figure 7 shows an object at two different representational levels. In a detailed representation, the object is composed of two holes and, in a more coarse representation, the two holes become only one. In both representations, however, the number of nodes and edges are the same.

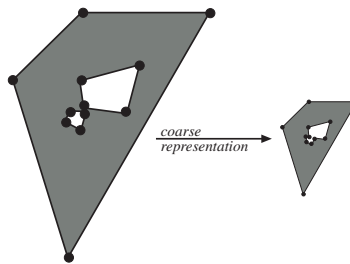
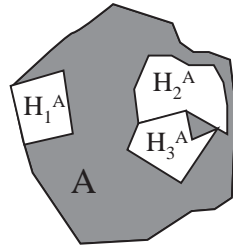


Fig. 7. Two representations of the same object

Considering objects' relations, Egenhofer *et al.* [24] present a framework that treats consistency at multiple representational levels based on the comparison of topological invariants [26]. They defined two types of equivalence: object equivalence and relation equivalence between different representations. This framework assumes that changes of topology through consecutive representational levels can be ordered by a similarity relation "topologically less general than or topologically as general as" (\leq), a relation that is reflexive, antisymmetric, and transitive. In this context, a representation is characterized by a set of topological invariants ($T(O_x^i)$) of an object (O_x) at a given representation (i), and a set of topological invariants between objects ($T(O_x^i, O_y^i)$) at a given representation (i).

The set of topological invariants of an object A ($T(A)$) is described by the relation matrix between the generalized object A^* (i.e., the object A without holes) and the object A 's holes H_i^A , and by the component invariant tables for the boundary-boundary intersections between holes and between the generalized object and the holes. The topological invariants of the boundary-boundary intersections include the sequence of intersections and the dimension of these intersections (i.e., zero-dimensional or point, one-dimensional or line). For example, Figure 8 shows a region A with three holes $H_1^A \dots H_3^A$, the relation matrix of the generalized region A^* and the holes H_i^A , and the component invariant tables for the boundary-boundary intersections. In this case, there is one boundary intersection between A^* and H_1^A and two boundary intersections between H_2^A and H_3^A .



	A^*	H_1^A	H_2^A	H_3^A
A^*	<i>equal</i>	<i>covers</i>	<i>contains</i>	<i>contains</i>
H_1^A	<i>covered by</i>	<i>equal</i>	<i>disjoint</i>	<i>disjoint</i>
H_2^A	<i>inside</i>	<i>disjoint</i>	<i>equal</i>	<i>meet</i>
H_3^A	<i>inside</i>	<i>disjoint</i>	<i>meet</i>	<i>equal</i>

$$TB(\delta A^*, H_1^A) = \begin{bmatrix} \text{sequence} & 1 \\ \text{dimension} & 1 \end{bmatrix}, TB(\delta H_2^A, H_3^A) = \begin{bmatrix} \text{sequence} & 1 & 2 \\ \text{dimension} & 1 & 0 \end{bmatrix}$$

Fig. 8. A region A with three holes $H_1^A \dots H_3^A$, the relation matrix between the generalized region A^* and the component invariant tables for the boundary-boundary intersections

The topological invariants between objects is characterized by the relation matrix between objects and by the topological invariants of the boundary-boundary intersections between objects. These invariants are the sequence of intersections, dimension of the intersections, type of intersections (i.e., an intersection crosses into or out of an object), and boundedness of boundary-boundary intersections (i.e., whether or not the components of boundary intersection are inside of the union of objects). As an example, consider the configuration in Figure 9 with two objects at a given representational level and their corresponding relation table and component invariant table of their boundaries.

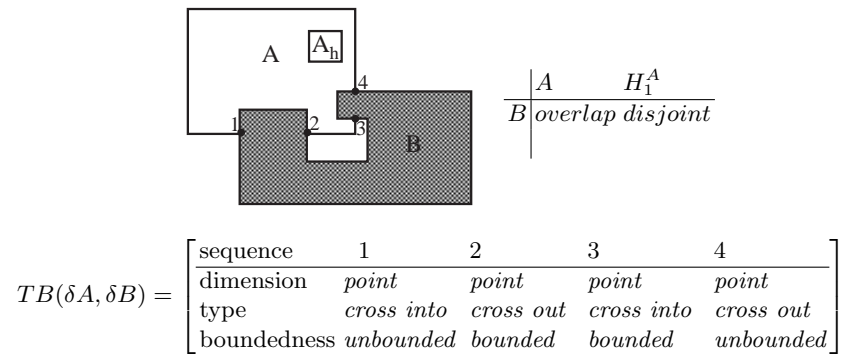


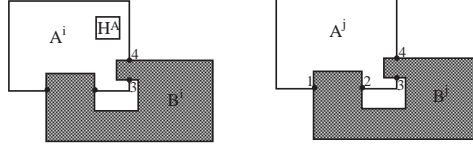
Fig. 9. A configuration with two objects and its corresponding relation table and component invariant tables

Egenhofer *et al.* classify the set of topological equivalences between representations into three types of similarity and three types of homeomorphism (Table 2). Within this framework, two representational levels are topologically consistent if they satisfied the conditions of topological homeomorphism; that is, if they have the same topological invariants and relation matrices. Two different representational levels may also be consistent if they satisfy some basic conditions of topological similarity (\leq) from a coarse to a detailed representation. The basic assumption when defining these consistency rules is that the goal of a coarse representation is to reduce the complexity of objects. For objects with holes, this means that the number of holes should be reduced in a coarse representation. Likewise, the number of boundary-boundary intersections between holes and between a generalized region and a hole should get smaller. If the topological relation between holes change, it changes from *disjoint* to *meet*. Thus, the dimension may increase from one to another representation. For example, if two holes are moved closer to each other, a component intersection that meets in a node may change to a meet in an edge. Like these basic rules, many other rules exist for objects with holes and for relations between objects with holes in different representations, which can be found in [24].

Table 2. Types of equivalence between representations S_i and S_j

Type	Rule
<i>object-similar</i>	$\forall(O_x^i \in S_i, O_x^j \in S_j)$ $[S_i \leq S_j \supset T(O_x^i) \leq T(O_x^j)]$
<i>relation-similar</i>	$\forall(O_x^i, O_y^i \in S_i, O_x^j, O_y^j \in S_j)$ $[S_i \leq S_j \supset T(O_x^i, O_y^i) \leq T(O_x^j, O_y^j)]$
<i>similar</i>	$\forall(O_x^i, O_y^i \in S_i, O_x^j, O_y^j \in S_j)$ $[S_i \leq S_j \supset (T(O_x^i, O_y^i) \leq T(O_x^j, O_y^j)) \wedge$ $(T(O_x^i) \leq T(O_x^j)) \wedge (T(O_y^i) \leq T(O_y^j))]$
<i>object-homeomorphic</i>	$\forall(O_x^i \in S_i, O_x^j \in S_j)$ $[S_i \leq S_j \supset T(O_x^i) = T(O_x^j)]$
<i>relation-homeomorphic</i>	$\forall(O_x^i, O_y^i \in S_i, O_x^j, O_y^j \in S_j)$ $[S_i \leq S_j \supset T(O_x^i, O_y^i) = T(O_x^j, O_y^j)]$
<i>homeomorphic</i>	$\forall(O_x^i, O_y^i \in S_i, O_x^j, O_y^j \in S_j)$ $[S_i \leq S_j \supset (T(O_x^i, O_y^i) = T(O_x^j, O_y^j)) \wedge$ $(T(O_x^i) = T(O_x^j)) \wedge (T(O_y^i) = T(O_y^j))]$

Figure 10 shows a case of *relation* homomorphism, because both representations have the same relation matrices, except for the fact that in the representation j a disjoint relation between objects B^i and H^A is dropped; and the representations have identical component invariant tables for the relations between non-empty boundaries intersections (i.e., $T(A^i, B^i) = T(A^j, B^j)$). In this case, the representation j may be considered a coarse or less detailed representation than the representation i .

**Fig. 10.** Two relation-homeomorphic representations

While the work by Egenhofer *et al.* [24] addresses consistency at multiple representational levels of objects with holes, a work by Tryfona and Egenhofer [63] focuses on the computational assessment of topological consistency across multiple representational levels of objects with disconnected parts. They define that the *generalized* region A^* of an object A with disconnected parts A_i is the union of all its parts and all relevant connectors ΔA_{ij} between parts A_i and A_j . A connector ΔA_{ij} between parts A_i and A_j is the region that links A_i and A_j , filling the exterior between the two parts such that A_i , ΔA_{ij} , and A_j are connected. The basic rules of the *generalized* object A^* and the parts A_i are:

$$\forall(i)[A^* \text{ covers } A_i] \quad (4)$$

$$\forall(i \neq j)[A_i \text{ disjoint } A_j] \quad (5)$$

$$\forall(i \neq j)[A_i \text{ meet } \Delta A_{ij} \wedge A_j \text{ meet } \Delta A_{ij}] \quad (6)$$

The goal of Tryfona and Egenhofer's work was to determine the relation between the generalized object A^* and another object B from the relations between B and A 's parts. This derivation is based on the analysis of the topological invariants defined by the set intersections of interior, boundary and exterior of objects [26], and on the consistency-checking of scenes [30]. For example, consider the case of two disjoint parts A_0 and A_1 and a third object B that contains A_1 (Figure 11). Then, a unique possible relation of the generalized object A^* with respect to B is *overlap*. A constraint of a relation between an object B and A^* is terms of an A 's part A_i can be expressed by:

$$\forall(A^*, B)[\text{overlap}(A^*, B) \equiv \exists(A_i)[\text{overlap}(A_i, B)]] \quad (7)$$

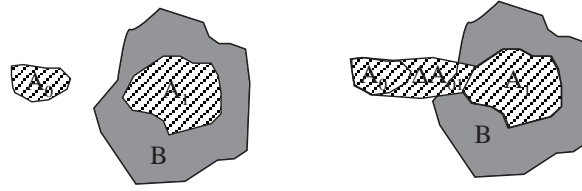


Fig. 11. Derivable relation between an aggregate object A and object B

In summary, multiple representations in spatial databases may not imply inconsistent information, but rather, merely different levels of detail or scale. In such cases, topological consistency at the level of objects and objects' relations must be analyzed. Analyses of consistency at multiple representational levels are not included in current commercial DBMS, they are running as ad-hoc applications. From the perspective of consistency in spatial databases, models of consistency at multiple representational levels lack the specification in a formal language for their treatments as integrity constraints.

4.3 Consistency in Spatial Information Integration

This Section discusses consistency in spatial information integration that considers cases where spatial data sets to be integrated contain the same features or objects, which can be extracted from several sources at different times. The

treatment of consistency when integrating data sets with different features (e.g., combining cadastral with water resource data) depends on the semantics of the features involved.

The integration of the same features from different sources may vary in reliability, accuracy and scale of representation. Thus, integrating spatial information may create conflicts due to the different representations for the same features concerning, for example, shape, dimension, and positional accuracy. As example, Figure 12 shows two objects, A and B , with different representations at the same representational level, each coming from a different source. The example in Figure 12 is inconsistent with respect to a constraint that specifies that objects must have only one geometric representation.

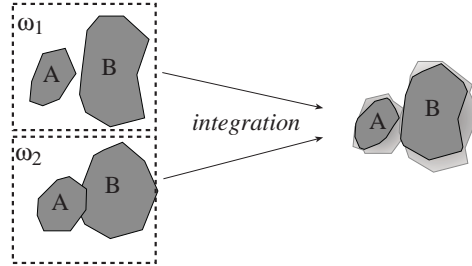


Fig. 12. Different representations of two objects

In the context of data integration, different types of consistency at the same representational level are distinguished [1]:

- *Total consistency* occurs when two configurations or data sets (i.e., when elements or objects that compose configurations) are identical.
- *Partial consistency* occurs when partial configurations are identical (i.e., when subsets of elements that compose configurations are identical).

As mentioned in Section 4.2 of consistency at multiple representational levels, two aspects of consistency when comparing data sets are object-based equivalence and relation-based equivalence. Object-based equivalence analyzes objects individually, so that it is possible to classify types of consistency in terms of the existence, shape, dimension, size, and degree of detail of objects. Relation-based equivalence focuses on objects' relationships, which are classified into topological, directional, or relative size equivalence.

The common approach to integrating different representations has assumed that when no further information exists about the origin of data, both representations are considered to equally contribute to the integration of information. In cases of multiple representational levels, a preliminary step is to check whether

or not different representational levels are consistent. When representational levels are consistent, a more detailed level can be mapped onto and integrated into a less detailed level; that is, into a representation generated by a generalization process. If, at a common representational level, two different representations exist, partial consistency may still be possible (i.e., parts of the different representations of an object or configuration are identical). The idea is to merge both representations in such a way that the resulting representation is modeled as a vague or unclear one. In modeling these unclear boundaries, three alternatives are found [24]:

- *Fuzzy models* [2][56][64][71], which are based on fuzzy set theory and have been applied to spatial uncertainty. Fuzzy set theory is an extension of classical boolean set theory that deals with different degrees of possibility that an individual is a member of a set or that a given statement is true [70]. Examples of fuzzy spatial objects are mountains, cities, and oceans.
- *Probabilistic models* [12][33], which are based on probability theory to model positional and measurement uncertainty. Probabilistic approaches model uncertainty by determining a degree of membership of an entity in a set in terms of statistically defined functions. An example of an unclear boundary that can be modeled by a probabilistic model is the water level of a lake that is not certainly known.
- *Exact models* [15][16][21][32], which map data models for spatial objects with sharp boundaries onto spatial objects with broad boundaries.

Consider an example of a fuzzy representation of indeterminate regions (Figure 13). A membership function for area A can be specified in 8, where B stands for the region that is definitely outside of A , A/B is a region that can be part of A or B , and d_a and d_b are the distances from a point (x, y) in the region A/B to the core area of the region A (i.e. region where $\mu_A(x, u) = 1$) and the core area of the region B (i.e., region where $\mu_A(x, u) = 0$):

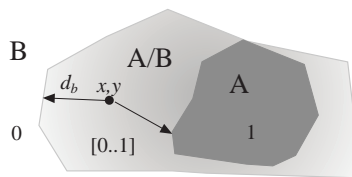


Fig. 13. A fuzzy region

$$\mu_A(x, y) = \begin{cases} 1 & \text{if } (x, y) \in A \wedge (x, y) \notin B \\ 1 - d_a / (d_a + d_b) & \text{if } (x, y) \in A \wedge (x, y) \in B \\ 0 & \text{if } (x, y) \notin A \wedge (x, y) \in B \end{cases} \quad (8)$$

In the context of data integration, fuzzy theory can be used in the integration of two representations that overlap (i.e., a partial consistency). In such case, one could consider that the overlapping areas or intersections between objects from different representations are the core areas of the integrated objects (i.e., dark grey of Figure 13 with membership function equal to 1) and the differences between the union and the intersection of representations are the unclear boundary of objects (i.e., light grey region with membership function in the range $[0 \dots 1]$). The regions outside of objects in both representations are considered outside of the integrated objects.

Following an exact approach to handle indeterminate boundaries, a broad boundary is associated with objects whose boundaries are unclear after integration. In the example of Figure 14, regions of broad boundaries are the regions that result from the difference between the union and intersection of objects in both representations, that is, the regions that do not clearly belong to the geometric representations of objects.

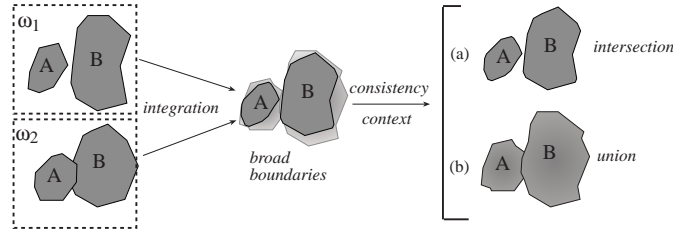


Fig. 14. Example of integration of spatial data based on previous knowledge of datasets

There are various possible strategies that can formalize the integration of more than one observation about location in a region with broad boundaries. These strategies make distinctions depending on the contextual information that characterizes the quality of representations [69]. For example, consider configurations in Figure 14, and assume that we know that both configurations (i.e., ω_1 and ω_2) are not accurate (i.e., there exist errors in positional information) then, only the intersection of both representations can be considered consistent (i.e., option (a) in Figure 14). If we consider that configuration ω_2 does not include regions that it should (i.e., incomplete representation), the union of both representations is considered consistent (i.e., option (b) in Figure 14).

Focusing on the integration of topological relations, the relation between objects with broad boundaries are described by an intersection matrix between in-

teriors (\circ) (i.e., between the core of objects), broad boundaries (Δ) (i.e., between the unclear regions of objects), and exteriors ($-$) [15] [60][67][68] (Figure 15). For topological relationships between regions with broad boundaries, 44 realizable matrices are possible.

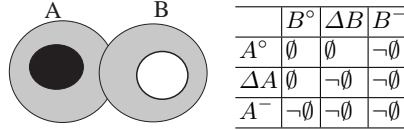


Fig. 15. Intersection matrix of objects with broad boundaries

Each intersection matrix of objects with broad boundaries has a set of topological relations that are realizable when considering changes from the core to the broad boundary of an object. In the case of Figure 15, three possible relations are realizable: *disjoint*, *meet* and *overlap*. Figure 16 shows these three alternatives when one considers that the geometry of objects change from the core to the broad boundaries. In this figure, gray lines represent the core boundary and broad boundary of objects.

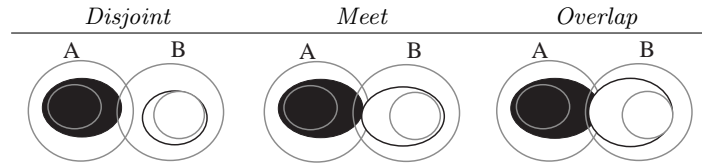


Fig. 16. Possible relations between objects with broad boundaries

To discuss the integration of different representations, consider the example in Figure 17, where two representations of two objects are integrated, resulting in objects with broad boundaries. The intersection of both representations define the core of objects and the difference between the union and the intersection defines the broad boundaries for each object. The idea here is not to analyze what the boundaries of individual objects are, but the relationship between objects; that is, what relationships are possible between the objects given that the integration of two representations results in objects with broad boundaries. From the point of view of consistency, if the analysis of broad boundaries determines that there exists only one possible relation between objects based on two representations, there is no conflicting information about the spatial relation between objects even in presence of different representations of objects. In

presence of multiple possible relations and a constraint that enforces a unique relation between objects, multiple representations are inconsistent. In the example of Figure 17, the broad boundaries (Δ) make possible that objects are *disjoint*, *meet* or *overlap*, that is, consistency cannot be guaranteed based on both representations.

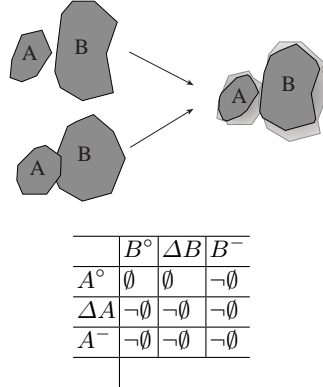


Fig. 17. Integration of two representations

In the same way than models for handling multiple representational levels of spatial objects, applications that integrate spatial information run as ad-hoc implementations, that is, they are user-defined applications rather than tools incorporated into current DBMS. These models also lack the specification in a formal language for their treatments as integrity constraints.

5 Consistency Tolerance in Spatial Databases

Although there has been active research on creating efficient spatial databases, the treatment of inconsistency in spatial databases is still a problem for current spatial information systems [8] [23]. The models described in the previous Sections about consistency at multiple representational levels and for data integration can be used in defining strategies for treating inconsistency in spatial databases; however, these models have not been integrated into a query process that explicitly addresses the answer and process of data despite the fact that the data are inconsistent (i.e., inconsistency tolerance). Inconsistency tolerance can be used to one's advantage when accessing or integrating data from different sources, or when it is inconvenient or impractical to enforce integrity constraints during data updates.

In traditional databases, studies have addressed inconsistency tolerance in query answers [3] [6] of a relational database schema with a set of integrity

constraints over this schema. From these studies, possible alternatives for dealing with inconsistency in query answering are: *ignoring inconsistency* (i.e., using conflicting data in the answer), *eliminating inconsistency* data (i.e., considering none of the conflicting data for answer; data cleaning), and considering the *consistent answer that belongs to all consistent states* of the database based on *minimum repairs*. Conceptually, these alternatives could be applied in the spatial domain when issuing queries that rely on spatial operations from the relational algebra with spatial criteria.

For example, consider a spatial database that has conflicting representations of spatial objects (conflicting representation of object A in Figure 18).

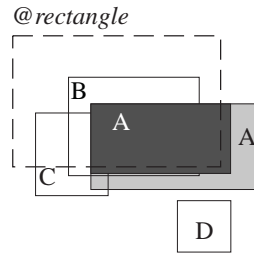


Fig. 18. An inconsistent database with different representations of an object A and a query defined by a window $@rectangle$

Using an extended relational database, the data set in Figure 18 is represented by a relation *Spatial Object* with the following instances, where $region_i$ represents a value of the geometric primitive *region*:

<i>Spatial Object</i>	<i>Name</i>	<i>Region</i>
	A	$region_1$
	A	$region_2$
	B	$region_3$
	C	$region_4$
	D	$region_5$

Having the functional dependency $Name \rightarrow Region$, meaning that the *Name* functionally determines *Region*, the relation *Spatial Object* violates the functional dependency, since there are two tuples with value A in attribute *Name*. In this example, two selection queries based on a space window (i.e., a rectangular area of the space $@rectangle$) that was defined by a user are:

```

1  select  clipping(r.geometry, @rectangle)
   from    Regions r
   where   Overlaps(r.geometry, @rectangle)

2  select  r.geometry
   from    Regions r
   where   Overlaps(r.geometry, @rectangle)

```

The first query returns the geometric parts of objects that overlap the window ($@rectangle$). The second query, on the other hand, returns the complete geometry of objects that totally or partially overlap the window. In answering these queries, the three alternatives of ignoring, eliminating, and considering minimum repairs of traditional databases can be applied (Figure 19). For the first query, ignoring inconsistency will return the geometric parts of both consistent and inconsistent data that overlap the query window. In this case, the conflicting representations of object A lay outside the overlapping region with the query window such that the answer does not have conflicting information. The situation is different, however, in the option of ignoring inconsistency for the second query, since the answer in that case will contain conflicting information, that is, two representations of object A that partially overlap the query window. In both queries, eliminating inconsistency data will not consider the conflicting data, that is, the geometry of object A is not considered as part of the answer.

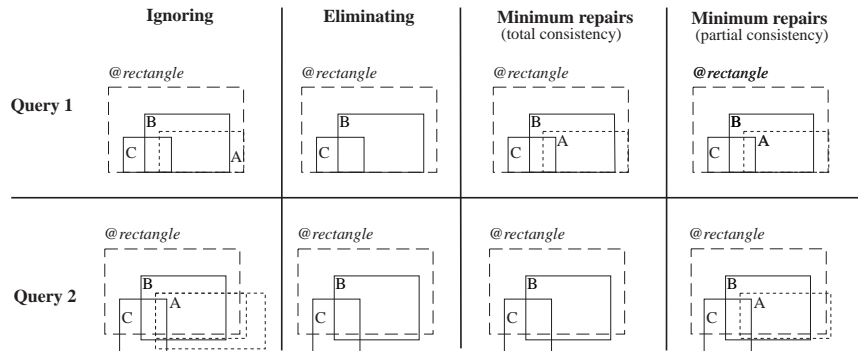


Fig. 19. Alternatives to consistent query answers from the inconsistent database

The option of minimum repairs returns all the answers that belong to the result of query evaluation in every repair [6]. In this database, there are two possible repairs, each of them considering only one tuple of object A . In answering the first query in Figure 19, both repairs contain the same geometric area of objects A , B , and C that overlap the query window (Figure 20); therefore,

these areas are part of the consistent answer. For answering the second query, since it is the complete geometry of objects that overlap the query window what is retrieved, only objects B and C can be considered consistent answers to this query.

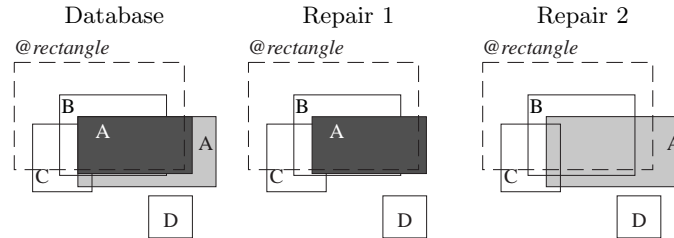


Fig. 20. Minimum repairs of the database

The previous example of the second query illustrates that in the case of minimum repairs, one could consider two further alternatives that depend on the granularity of the determination of inconsistency in the geometric representation of spatial objects. One option is to take the geometry of an object as a whole, which is the basic case described before when making the repairs based on the complete geometries of objects (Figure 20). The second alternative of minimum repairs is to consider that the geometry of an object can be partially inconsistent, in which case, the repair of the database takes the consistent parts of objects' representations. Partial consistency may be defined by the part of the geometric representation of objects that is equivalent in conflicting information, that is, the intersection of geometric representations. For example, in the previous database, one of the representations of object A is inside of the other such that the former corresponds to the intersection of both representations. This intersection region, by definition, is present in all representations of object A and, therefore, it is part of the repair (Figure 21). In the example of the queries in Figure 19, the intersection of the geometric attribute of both tuples with value A in attribute *Name* will be considered in the answer.

Other interesting queries are those that uses criteria defined by spatial relations between objects. Spatial relations are usually derived during the query process, and they may not require accurate data about positional information. This type of query involves spatial joins between relations, which construct the pairs of tuples from the relations whose spatial components satisfy spatial predicates. When querying by spatial relations between objects, inconsistency with respect to the representation of objects (i.e., location and shape of objects) may not affect the consistency with respect to the spatial relation between objects (e.g., even with conflicting positional information exists, objects can still hold the same topological relation). What is more, by considering some metric refinements of topological relations (i.e., relative size and distance of objects) [36], one

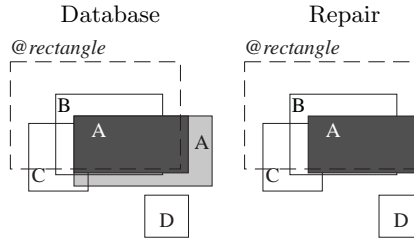


Fig. 21. The minimum repair of the database that considers partial consistency

could also determine that objects keep the same topological relations despite the fact that they have conflicting geometric representations.

Consider the same database of spatial objects in Figure 18 and the following two selection queries based on a criteria of spatial relations between objects, where the difference is in the selection component (i.e., regions' ids or regions' geometries):

- 1 **select** $r_1.id$
 from $Regions\ r_1, Regions\ r_2$
 where $r_2.id = 'B'$ and $Overlaps(r_1.geometry, r_2.geometry)$

- 2 **select** $r_1.geometry$
 from $Regions\ r_1, Regions\ r_2$
 where $r_2.id = 'B'$ and $Overlaps(r_1.geometry, r_2.geometry)$

In answering these queries, the geometric representation is needed for applying the spatial criteria. The answer to the first query, however, does not concern the selection of the geometric representation of objects, but the selection of objects' ids. Thus, since the spatial criteria is satisfied in both representations of A , the answer to the query is the same in all repairs of the database. For the second query, in contrast, the answer is the geometric representation of objects so that, even if the spatial criteria is satisfied in all repairs of the database, the answer cannot include conflicting information (Figure 22).




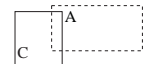
	Ignoring	Eliminating	Minimum repairs (total consistency)	Minimum repairs (partial consistency)
Query 1	A,C	C	A,C	A,C
Query 2				

Fig. 22. Alternatives to consistent query answers from the inconsistent database

A different perspective for handling conflicting representations in answering a query is the use of broad boundaries in a query concerning the geometric aggregation of objects. This idea of objects with broad boundaries could be related to the way aggregate functions have been treated in consistent query answering from inconsistent traditional databases [4] [5] [14]. A consistent answer to an aggregation query is defined as a *minimum interval* such that the value of the aggregation function in every repair of the database belongs to this *minimum interval*. The end-points of the *minimum interval* corresponds to the greatest lower bound and the least upper bound answers to the query in the database.

In a query by aggregation of objects' geometry, two different geometric representations may lead to different aggregate objects. Answering a query may then involve treating the aggregate object as an object with broad boundaries, that is, an object with a crisp boundary defined by the intersection of all possible aggregations, and with a broad boundary defined by the union of all aggregations (Figure 23). In this case, the *minimum interval* of possible answer are limited by the greatest lower bound that corresponds to the crisp region and the least upper bound that corresponds to the region defined by the broad boundary.

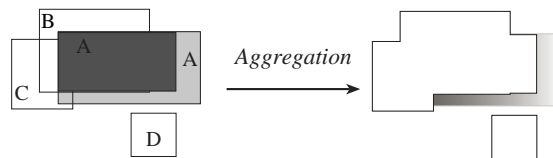


Fig. 23. Aggregation of inconsistency representations by modeling objects with broad boundary

The previous examples have described conceptually the use of different alternatives for dealing with inconsistency tolerance with respect to geometric representations, without taking in consideration the computational mechanisms for obtaining consistent answers. Other examples with different types of inconsistencies, such as semantic inconsistency and topological inconsistency, and more complex queries are also possible. In all cases, much work need to be done with respect to what repairs and consistent answers are in spatial databases.

The treatment of inconsistency tolerance raises new issues respect to topological constraints. In such cases, the inconsistency is not the result of conflicting information about the position of objects, but rather of a lack of consistency with the rules that define the primitives of representation. For example, a typical structural constraint of a polygon is to be bounded by a non self-intersecting and closed polyline. The satisfaction of topological constraints ensures that some computational-geometry algorithms can be successfully executed; however, not all of these algorithms require the same structural constraints. For example, the boundary of a region must be defined by a closed polyline in order to calculate the area of a region. On the other hand, a closed and non self-intersecting polyline is the requirement of an algorithm for determining whether or not a point is inside of a region. So, if one only wants to compute the area of a polygon, polylines only need to be closed. This analysis may imply that topological constraints may be associated with the particular use of spatial operators rather than with a general definition of geometric primitives.

6 Conclusions

This chapter presents a review of the work on inconsistency in spatial databases. It discusses the kinds and origins of inconsistency, the specification of integrity constraints, and the treatment of inconsistency for representations at different levels and for data integration. Further, it discusses how inconsistency tolerance can be introduced in querying inconsistent spatial databases. This review highlights issues about composite objects and spatial relations in the treatment of inconsistency.

Summarizing, important issues for the treatment of inconsistency that were discussed are:

- Inconsistency may relate to conflicting information with respect to positional or qualitative attributes of objects or to contradictions with respect to structural and semantic rules. Structural conditions of geometric primitives have been typically expressed as integrity constraints.
- Integrity constraints may refer to the geometric representation of objects by making reference to conditions on the geometric types of objects (i.e., point, polylines, regions), or they may refer to the semantic of spatial objects (e.g., a road cannot run into a body of water). Thus, constraints can be expressed, for example, by points or aggregations of points (i.e, by geometric primitives), or by objects or aggregations of objects (i.e., by objects that have a semantic meaning, such as rivers, building and roads).

- Queries concerning spatial relations may not need a unique geometric representation of objects and, therefore, such queries are less sensitive to conflicting positional information.
- Definitions of composite objects deal with sets of objects and impose constraints between wholes and parts to enforce consistency.
- Geometric information about spatial objects can be considered as a whole unit (i.e., a geometric representation is consistent or inconsistent as a whole) or can be considered as an aggregation of spatial parts (i.e., a geometric representation is consistent, partially consistent, or totally inconsistent).
- Multiple representation levels with respect to different scales may be necessary in information systems. In such cases, multiple representations are considered consistent if they satisfy basic topological constraints.
- Different definitions of consistent answers and database repairs can be applied to spatial databases based on the interpretation and use of the geometric representations.

Since this chapter has outlined issues concerning the treatment of inconsistency of spatial databases, it leaves the door open for exploring aspects of formalization and implementation of mechanisms for consistent query answering from inconsistent spatial databases. Although it was not discussed in the chapter, there is an increasing interest in the research community of spatial databases toward the management of spatial-temporal applications. These types of applications raise issues of temporal, spatial, and spatial-temporal consistency [40] [50] [51].

Acknowledgment. Andrea Rodríguez’s research work is partially funded by CONICYT under grant FONDECYT 1030301.

References

1. A. Abdelmoty and C. Jones. Towards maintaining consistency in spatial databases. In *Proceedings of the Sixth International Conference on Information and Knowledge Management*, pages 293–300, Las Vegas, USA, 1997. ACM Press.
2. D. Altman. Fuzzy set theory approaches for handling imprecision in spatial analysis. *International Journal of Geographic Information Science*, 8(3):271–289, 1994.
3. M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 68–79, 1999.
4. M. Arenas, L. Bertossi, and J. Chomicki. Specifying and querying database repairs using logic programs with exceptions. In *International Conference on Flexible Query Answering Systems (FGAS)*, pages 27–41. Springer-Verlag, 2000.
5. M. Arenas, L. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming*, 3(4 & 5):393–424, 2003.
6. L. Bertossi and J. Chomicki. Query answering in inconsistent databases. In J. Chomicki, G. Saake, and R. van der Meyden, editors, *Logics for Emerging Applications of Databases*. Springer-Verlag, 2003.

7. P. Bonnisson and R. Tong. Reasoning with uncertainty in expert systems. *International Journal of Man and Machine Studies*, 22:241–250, 1985.
8. K. Borges, C. Davis, and A. Laender. Integrity constraints in spatial databases. In *Database Integrity: Challenges and Solutions*. Ideas Group, 2002.
9. K. Borges, A. Laender, and C. Davis. Spatial integrity constraints in object oriented geographic data modeling. In C. Bauzer-Medeiros, editor, *ACM International Symposium on Advances in GIS*, pages 1–6. ACM Press, 1999.
10. P. Bosc and H. Prade. An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management systems. In A. Motro and P. Smets, editors, *Uncertainty Management in Information Systems: From Needs to Solutions*, pages 285–324. Kluwer Academic Publishers, 1997.
11. T. Bruns and M. Egenhofer. Similarity of spatial scenes. In *International Symposium on Spatial Data Handling SDH'96*, pages 31–42, Delf, The Netherlands, 1996. Taylor and Francis.
12. P. Burrough. Natural objects with undeterminate boundaries. In A. Frank, editor, *Geographic Objects with Indeterminate Boundaries GISDATA*, pages 3–28. Taylor & Francis, 1996.
13. A. Chandra and D. Harel. Computable queries for relational database systems. *Journal of Computer and System Sciences*, 21(2):156–178, 1980.
14. J. Chomicki. Consistent query answering: Recent developments and future directions. In S. Jajodia and L. Strous, editors, *Working Conference on Integrity and Internal Control in Information Systems*, Lousanne, Switzerland, 2003. Kluwer Publishers.
15. E. Clementini and P. Di Felice. An algebraic model for spatial objects with undeterminate boundaries. In A. Frank, editor, *Geographic Objects with Indeterminate Boundaries*, pages 155–169, London, 1996. Taylor & Francis.
16. E. Clementini and P. Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16:73–204, 1997.
17. E. Clementini, J. Sharma, and M. Egenhofer. Modeling topological relations: Strategies for query preprocessing. *Computers and Graphics*, 18(6):815–822, 1994.
18. S. Cockcroft. A taxonomy of spatial integrity constraints. *GeoInformatica*, 1(4):327–343, 1997.
19. S. Cockcroft. Modelling spatial data integrity constraints at the metadata level. In D. Pullar, editor, *GeoComputation*, Brisbane, Australia, 2001.
20. A. Cohn, B. Bennett, J. Gooday, and N. Gotts. Representing and reasoning with qualitative spatial relations about regions. In O. Stock, editor, *Spatial and Temporal Reasoning*, pages 97–134. Kluwer Academic Publishers, 1997.
21. G. Cohn and N. Gotts. The 'egg-yolk' representation of regions with indeterminate boundaries. In A. Frank, editor, *Geographic Objects with Indeterminate Boundaries*, pages 171–187, London, 1996. Taylor & Francis.
22. H. Couclelis. People manipulate objects (but cultivate fields): Beyond the raster-vector debate in gis. In A. Frank, I. Campari, and U. Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. LNCS vol. 639*, pages 65–77. Springer-Verlag, 1992.
23. M. Egenhofer. Consistency revisited. *GeoInformatica*, 1(4):323–325, 1997.
24. M. Egenhofer, E. Clementini, and P. Di Felice. Evaluating inconsistency among multiple representations. In *Spatial Data Handling*, pages 901–920, Edinburgh, Scotland, 1994.
25. M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

26. M. Egenhofer and R. Franzosa. On the equivalence of topological relations. *International Journal of Geographical Information Systems*, 8(6):133–152, 1994.
27. M. Egenhofer and J. Herring. Categorizing topological spatial relations between point, line, and area objects. Technical Report Report 94-1, National Center for Geographic Information Analysis, 1994.
28. M. Egenhofer and D. Mark. Naive geography. In A. Frank and W. Kuhn, editors, *Theoretical Basis for Geographic Information Systems COSIT'95*, pages 1–14, Semmering, Austria, 1995. Springer-Verlag.
29. M. Egenhofer and J. Sharma. Topological consistency. In P. Bresnahan, E. Corwin, and D. Cowen, editors, *Proceedings of the 5th International Symposium on Spatial Data Handling*, pages 335–343, Charleston, USA, 1992. IGU Commission of GIS.
30. M. Egenhofer and J. Sharma. Assessing the consistency of complete and incomplete topological information. *Geographical Systems*, 1(1):47–68, 1993.
31. R. Elsmari and S. Navathe. *Fundamentals of Database Systems*. Addison Wesley, 3er edition edition, 2000.
32. M. Erwing and M. Schneider. Vague regions. In *5th Int. Symposium on Advances in Spatial Databases SSD97. LNCS 1262*, pages 298–320. Springer-Verlag, 1997.
33. J. Finn. Use of the average mutual information index in evaluating error and consistency. *International Journal of Geographic Information Science*, 7(4):349–366, 1993.
34. A. Frank. Tiers of ontology and consistency constraints in geographical information systems. *International Journal of Geographic Information Science*, 15(7):667–678, 2001.
35. G. Gardarin, J.P. Cheiney, G. Kiernan, D. Pastre, and H. Stora. Managing complex objects in an extensible relational DBMS. In *Proceedings of Very Large Data Bases*, 1989.
36. F. Gody and A. Rodríguez. A quantitative description of spatial configurations. In D. Richardson and P. van Oosterom, editors, *Proceedings of the 10th Symposium on Spatial Data Handling*, pages 299–311. Springer, 2002.
37. R. Güting. An introduction to spatial database systems. *VLDB Journal*, 3:357–399, 1994.
38. R. Güting and M. Schneider. *International Symposium on Spatial Databases, Lecture Notes in Computer Science Vol. 692*, chapter Realms: A Foundation for Spatial Data Types in Database Systems, pages 14–35. Springer-Verlag, 1993.
39. R. Güting and M. Schneider. Realm-based spatial data types: The rose algebra. Technical Report 141-3-93, Fern Universität, Hagen, 1993.
40. R. Güting, M. Böhlen, M. Erwing, C. Jensen, N. Lorentzos, M. Schneider, and M. Vazirgiannis. Foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25(1):1–42, 2000.
41. T. Hadzilacos and N. Tryfona. A model for expressing topological constraints in geographic databases. In A. Frank, I. Campari, and U. Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space COSIT92*, pages 252–268, Pisa, Italy, 1992. Springer-Verlag.
42. P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51(1):26–52, 1995.
43. B. Kuipers, J. Paredaens, and J. den Busshe. On topological equivalence of spatial databases. In F. Afrati and Ph. Kolaitis, editors, *6th International Conference on Database Theory ICDT97, LNCS 1186*, pages 432–446. Springer Verlag, 1997.
44. G. Kuper, L. Libkin, and J. Paredaens. *Constraint Databases*. Springer-Verlag, 2000.

45. R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Academic Press, 1992.
46. D. Papadias and Y. Theodoridis. Spatial relations, minimum bounding rectangles and spatial data structures. *International Journal of Geographical Information Science*, 11(2):111–138, 1997.
47. J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial databases queries. In *Proceedings of the 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 128–288. ACM Press, 1994.
48. J. Paredaens and B. Kuipers. Data models and query languages for spatial databases. *Data Knowledge Engineering*, 25(1-2):29–53, 1998.
49. S. Parson. Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 8(3):353–371, 1996.
50. D. Pfoser and C. Jensen. Capturing the uncertainty of moving-object representations. In R. Güting, D. Papadias, and F. Lochovsky, editors, *Proceedings of the 6th International Symposium on the Advances in Spatial Databases*, pages 111–132. Springer Verlag, 1999.
51. D. Pfoser and N. Tryfona. Capturing fuzziness and uncertainty of spatiotemporal objects. In A. Caplinskas and J. Eder, editors, *5th East-European Conference on Advances in Databases and Information Systems ADBIS01. LNCS 2151*, pages 112–126, Lithuania, 2001. Springer Verlag.
52. A. Pizzaro, A. Klinger, and A. Cardenas. Specification of spatial integrity constraints in pictorial databases. *Computer*, 22(12):59–71, 1989.
53. L. Plümer and G. Gröger. Achieving integrity constraints in geographic information systems. *GeoInformatica*, 1(4):345–367, 1997.
54. P. Ragaux, M. Scholl, and A. Voisard. *Spatial Databases: with Application in GIS*. Academic Press, 2002.
55. H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.
56. M. Schneider. Metric operations on fuzzy spatial objects in databases. In *Proceedings of the 8th ACM Symposium on Geographic Information Systems*, pages 21–26, Washington DC, 2000. ACM Press.
57. S. Servige, T. Ubeda, A. Puricelli, and R. Laurini. A methodology for spatial consistency improvement of geographic databases. *GeoInformatica*, 4:7–24, 2000.
58. S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice Hall, 2003.
59. S. Shekhar, M. Coyle, B. Goyal, D.-R. Liu, and S. Sarkar. Data models in geographic information systems. *Comm. ACM*, 40(4):103–111, 1997.
60. J. Stell and M. Worboys. Stratified map spaces. In T. Poiker and N. Chrisman, editors, *Spatial Data Handling*, pages 180–189, British Columbia, Canada, 1998. Taylor & Francis.
61. O. Stock. *Spatial and Temporal Reasoning*. Kluwer Academic Publishers, 1997.
62. M. Stonebraker and L.A. Rowe. The design of POSTGRES. In *Proceedings of ACM SIGARCT-SIGMOD*, pages 340–355, 1986.
63. N. Tryfona and M. Egenhofer. Consistency among parts and aggregates: A computational model. *Transactions on GIS*, 1(3):189–206, 1997.
64. E. Usery. A conceptual framework and fuzzy set implementation for geographic features. In A. Frank, editor, *Geographic Objects with Undeterminate Boundaries GISDATA*, pages 71–85. Taylor & Francis, 1996.
65. A. Voisard and B. David. A database perspective on geospatial data modeling. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):226–246, 2002.

66. M. Worboys. A geometric model for planar geographical objects. *International Journal of Geographical Information Systems*, 6(5):353–372, 1992.
67. M. Worboys. Computation with imprecise geographic data. *Journal of Computers, Environment and Urban Systems*, 22:85–106, 1998.
68. M. Worboys. Imprecision in finite resolution spatial data. *GeoInformatica*, 2:257–280, 1998.
69. M. Worboys and E. Clementini. Integration of imperfect spatial information. *Journal of Visual Languages and Computing*, 12:61–80, 2001.
70. L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–358, 1965.
71. F. Zhan. Approximate analysis of binary topological relations between geographic regions with indeterminate boundaries. *Soft Computing*, 2:28–34, 1988.