Deduction and Deductive Databases for Geographic Data Handling

A. I. Abdelmoty, M. H. Williams and N. W. Paton

Department of Computing and Electrical Engineering, Heriot-Watt
University, Riccarton, Edinburgh EH14 4AS, United Kingdom

Abstract. The representation of complex spatial domains in conventional databases suffers from fragmented representation of object structure, lack of instance-level spatial relationships, and the generation of large combinatoric search spaces in query analysis. The deductive capabilities provided by a deductive database offer some assistance in solving these problems, in particular by enabling spatial reasoning to be performed by a Geographic Information System (GIS). Deduction in the database is used to support the natural representation of complex spatial object structures in single and multi-layered Geographic DataBases (GDB), inference of implicit spatial relationships, and the manipulation of multiple resolution spatial representations. In addition, deductive capabilities are shown to be essential for automatic data input and update in a GDB. Coupled with appropriate structural representation, spatial reasoning is an important tool for the realization of an effective GDB.

1 Introduction

Complex real-world decision making tasks typically required in a GIS depend on a human's natural spatial, temporal and hierarchical reasoning ability. The automation or partial automation of such tasks depends heavily on search efficiency which in turn depends heavily on data modelling and representation. Modelling geographic data combines the complexity of modelling large spatial domains with the complexity incurred from the type of applications which have to be handled. This paper presents an investigation into the application of deduction in the context of databases to the representation and manipulation of geographic data with the aim of optimizing data storage and showing intelligent behaviour, which reflects itself in more efficient GISs.

Manipulating geographic data involves evaluation of spatial properties and relationships which necessitates the existence of a spatial model for data representation. This is usually achieved by viewing the geographic space as a collection of spatial entities such as points, lines, polygons or point sets, and representing objects and spatial relationships accordingly [27]. On the other hand, a different type of model is required to represent the aspatial aspects of geographic data. The hierarchical nature of the spatial data and the complex relationships to be represented have limited the use of the relational approach [7] and led towards modelling using an object-oriented representation [35, 9, 20]. However, using a single data model for representing both the spatial and aspatial aspects of geographic data meant forcing the spatial aspects of the data into the underlying data model whether relational [15] or object-oriented [33]. A model which is used to represent geographic data should ideally,

- Provide a rich set of semantic modelling capabilities to represent geographic entities as a coherent combination of spatial and aspatial aspects.
- Enable one to reason over the data in the same way as a pure spatial model would, viz. representation of complex spatial structures, multiple representation of geographic entities, and representation and inference of spatial relationships.

To this end we are currently investigating the application of the deductive object-oriented approach to databases to the realization of GDBs as a specialization of large spatial databases, taking into account the types of analysis and manipulation required in a GIS. Deduction in the database is a powerful mechanism for expressing queries, deriving data, and expressing integrity constraints, while object-orientation is appropriate for representing complex object structures and semantic relationships using concepts of data abstraction, encapsulation of structure and behaviour, and inheritance.

In this paper a comprehensive overview of the application of deduction in large GDBs is reported, covering different areas and levels in a GIS. It focuses on the representation of the spatial structure of a geographic object, spatial reasoning and spatial relationship inference, the definition of object classes over more than one data layer, and the choice of an appropriate spatial representation for a geographic object during query analysis.

A hypothetical geographic application based on resource management and allocation has been designed to test some of the issues discussed in this paper and will be used as test example for a prototype deductive object-oriented database system [12]. The application of a rule-based approach to feature extraction from digital maps has already been implemented using Prolog applied to data from Ordnance Survey large scale maps. Implementation issues are outside the scope of this paper and will be covered in future work.

The paper is organized as follows. Section 2 presents our view of a GDB as an object-oriented database with deductive capabilities. In section 3 the analysis of a GDB as a multidimensional framework for data is presented, pointing out areas where deduction in a database is applicable and useful. Expressing geographic database queries using a logic language is presented in section 4, while section 5 gives a general account of the application of deduction in automating the process of object recognition from input data sources in the form of maps and images.

2 Overview of a Deductive Object-Oriented Geographical Database

In this work a geographical database is regarded as consisting of two disjoint sets of database relationships, as shown in figure 1: one is the set of base or extensional database relationships (*EDB* relationships) and the other is the set of derived or intensional database relationships (*IDB* relationships). The *EDB* in turn contains two different levels of data representation, viz.

1. *Primitive level*, which is a spatial representation of geographic objects using an appropriate geometric data model [14], either in vector form (points, lines, and polygons) or tesselated (raster) form. Data from multiple sources is initially transformed and represented at this level. Spatial indexing structures are used to implement the geometric data model to improve performance, especially in search operations. Computational geometry algorithms necessary for spatial operations are also defined on the geographic data at this level.
2. *Object level*, where real world phenomena are represented as classes of objects which encapsulate their structure and behaviour. A rich set of semantic relationships including aggregation, specialization, association, etc. are used to represent complex abstract geographic phenomena. A mapping exists between the object level and the primitive one, i.e. there exists a function or a sequence of functions for every geographic object which leads to its spatial representation.

The *IDB* is the set of rules over objects in the *EDB*, which are used for spatial reasoning over the geographic space (as will be discussed later) and for feature extraction, whereby object level entities are inferred from the corresponding primitive primitive level objects. This is seen to be a necessary part of a GDB system for initial data loading and updating.

3 Deduction for Managing Geographic Database Dimensions

While knowledge of its shape and location in space might be enough to define a spatial entity, this is not sufficient for defining objects in a geographic space. This is due to the wide range of different applications that can be associated with the same area in space. There are as many different applications of geographic databases as there are kinds of maps or combinations of different kinds of maps.

A GIS user is interested not only in the extension of a particular object in space, but also in the different phenomena collected over a particular location in space, the ways in which these change over time, and the different representations of objects under different manipulation operations. By classifying the different kinds of maps used, a general view of the "dimensions" through which a geographic database has to extend can be obtained. Four different dimensions have been identified, namely, space, theme, resolution and time, as illustrated in figure 2. In this section a detailed definition of the first three dimensions is presented along with the effect of these dimensions on the representation of data in a GDB. This study is essential for pointing out particular areas in geographic data modelling which cannot be readily handled using a conventional DBMS and to show how the deductive capabilities of a database system can be used to handle some of these representational problems.

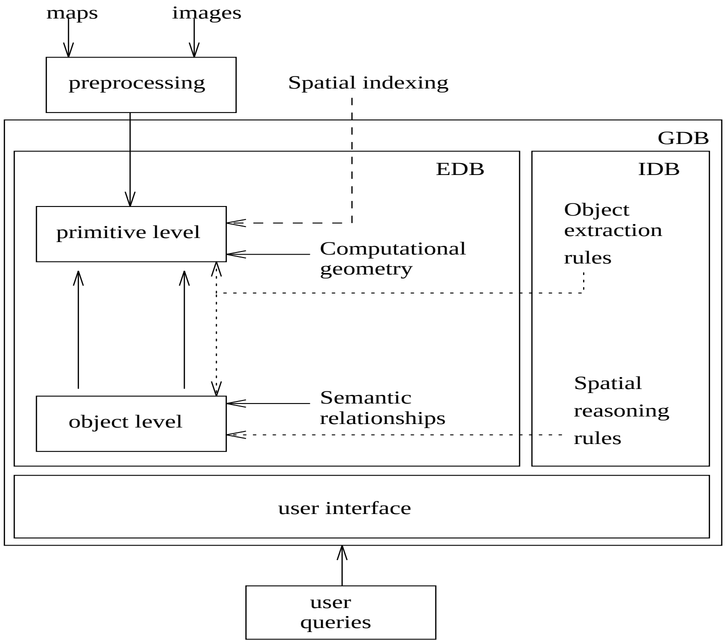

Fig. 1. Components of a deductive object-oriented GDB

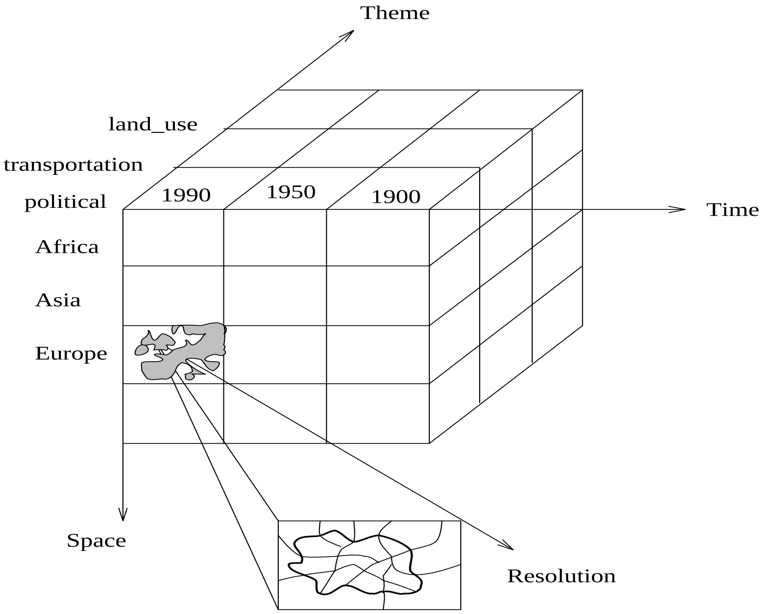

Fig. 2. Different dimensions of geographic data

3.1 Space: Representing Different Areas of Interest

The space dimension is concerned with the location and spatial extent of an object. A map is a two dimensional representation (x,y) of the real world, where location is described in terms of latitude and longitude. The third dimension (z) in space is represented explicitly using special types of maps, e.g. contour maps. Variation along this dimension represents different areas of interest. Existence in space affects geographic object definition in two ways: the definition of complex spatial objects and spatial relationships, as discussed below.

Complex Geographic Objects. The effect of the space dimension of geographical data is reflected in the classification of objects in a geographic database. In non-geographic application domains, objects tend to be grouped or classified on the basis of intrinsic resemblances or differences among instances. By contrast, in a geographic application domain, objects are required to be classified both on that basis and simultaneously on the basis of their constituting a spatial totality wherein the concepts of spatial proximity or separation are potentially just as meaningful as those of intrinsic likeness or unlikeness. Classification of objects on the basis of location almost always includes spatial relationships that the class of objects exhibits with respect to other classes. This characteristic is known as the *spatial structure* of geographic objects.

Thus the definition of an object could comprise a complex pattern of properties and spatial relationships, rather than a simple grouping of the related objects. The structure of the pattern is spatially defined as the location of each object relative to each other object in the pattern. This spatial structure cannot be explicitly defined in a relational database or an object-oriented database, but can be naturally defined using rules in a deductive database as follows.

Consider the definition of a road network object as a set of connected roads,

$$road\_network(road\_list) \leftarrow connected(road\_list).$$

The definition of a parish object can be formulated using the following rule, assuming that a county is divided into a set of non-overlapping parishes.

$$\begin{aligned} parish(x,y) \leftarrow\ & county(y) \wedge coveredby(x,y) \wedge parish(z,y) \\ \wedge\ & spatial(x, xpolygon) \wedge spatial(z, zpolygon) \\ \wedge\ & (disjoint(xpolygon, zpolygon) \vee meet(xpolygon, zpolygon)). \end{aligned}$$

where the disjoint, meet and coveredby relationships are as shown in figure 3. A complex geographic object is thus defined in terms of a particular view of other geographic objects and/or through specific relationships with those objects. The extension set of such objects (i.e. the instances of the object class) need not always be initially defined, as it could extend over the entire space limit of the database, while it is always the case that the instantiations of those object classes are required only for a particular map space. Thus the most appropriate way of defining these classes is through general rules in the *IDB*. To conduct any analysis procedure over a geographic database, the user has to specify the spatial limits of the data involved in the process (i.e. specify the area of interest

to which analysis has to be limited and the rules to be applied).
Rules defining a particular class of objects could change from one location to another for several reasons. In particular,

1. The shapes and properties of man-made objects can be different in different areas. For example, grain silos in the US are round, while those in Canada are square; buildings with associated area might be interpreted as garages in some parts of a city (too small for a dwelling) whereas in other parts of the city they may be taken as modern housing.

$$\begin{aligned} grainsilo(x) &\leftarrow building(x) \wedge round(x) \wedge \ldots \\ garage(x) &\leftarrow building(x) \wedge area(x, y) \wedge lessthan(y, a) \wedge \ldots \end{aligned}$$

2. A particular law may apply in one location and not in another. For example, laws governing the construction of houses, roads, etc. may differ from one area to another; parking laws may be different for different road and street types and so on.

$$\begin{aligned} parking\_law(x, metred\_parking) \leftarrow\ & road(x) \wedge road\_type(x, major) \\ & \wedge\ \neg day(saturday) \wedge \neg day(sunday) \\ & \wedge\ timeofday(`9 - 18`). \end{aligned}$$

Consequently some rules may be constrained over particular space limits in the database, with different versions of the rule defined for different spatial areas.

Spatial Relationships. In other application domains, relationships between objects are defined at the object class level. For example, *lecturer teaches course* and *student takes course*. In a geographic domain an object exhibits spatial relationships with all other objects in the database. Some spatial relationships are general (depending on the application, and on the spatial context) and apply to all objects within a class, whereas others are specific relationships between instances of geographic objects. In the latter case, a geographic object is considered to be in association or correlation with some other object or group of objects in space, which implies the relative description of its location. For example, objects are frequently used as landmarks to define locations of other objects, (the second house beside the church, the first street on the right after the national theater, and so on), or just expressing an ad hoc relationship between objects such as, city A is near lake B, or country C is in Western Europe.

This kind of object instance relationship is common in GIS queries. The representation and efficient retrieval of such relationships are essential functions of a GIS. It is not feasible to store all such relationships explicitly. Consequently, the dynamic evaluation of spatial relationships is necessary. On the other hand, it is not practical to specify explicitly the computations involved in the relationship every time a query is invoked. One way of supporting efficient computation involving spatial relationships is through special indexing structures, such as quad-trees, kd-trees, etc. [31, 25], over the primitive representation level of the geographic space to support different types of space analysis. Evaluation of relationships using this method requires the transition between the two different levels of representation, which can result in large computational overheads.

Alternatively, spatial reasoning can be applied over entities on the object level[1]. The identification, classification, and formal definition of spatial relationships is necessary for any spatial reasoning to be applied. Two specific frameworks for the representation of spatial relationships on spatial regions have been presented in Egenhofer[8] and Randell[30]. In what follows, both are reviewed and we show how such representation frameworks can be effectively implemented within a database using deductive rules.

Relationships Involving Regions.
(A) Egenhofer's Mathematical Model: A formal model for the combination of topological knowledge and the derivation of compositions of binary topological relationships is proposed by Egenhofer [8], based on a model for spatial data and relationship representation [29, 6, 11, 10, 8] which is based on concepts of point-set topology with open and closed sets [23]. Topological relationships between two point sets are defined through intersection relations of their boundary, interior and complement.

Reasoning over spatial relationships is then the composition of two binary relations over a common object i.e. R_3(a,c) can be derived from R_1(a,b) and R_2(b,c). An exhaustive set of 64 such compositions based on 8 relationships (**d**isjoint, **m**eet, **e**qual, **i**nside, **c**overed**by**, **c**on**t**ains, **c**o**v**ers, **o**verlap) between two point-sets have been defined in [8]. Figure 3 shows the representation of these relations.

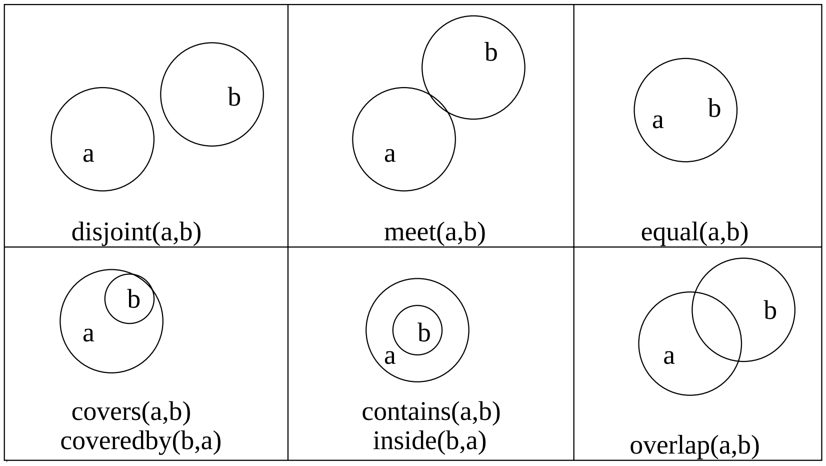

Fig. 3. Relationships between two regions in 2-dimensional space adapted from [8]

Such a representation framework can be implemented in a deductive database. The composition of spatial relationships can be rewritten as database clauses where the conjunction of two binary relationships R_1(a,b) and R_2(b,c) can be

---

[1] *Spatial reasoning* is the method by which spatial information which has not been explicitly recorded can be deduced.

mapped to a disjunctive set of base relations, for example,

$$inside(a,c) \leftarrow inside(a,b) \wedge inside(b,c). \tag{1}$$

$$\begin{aligned} disjoint(a,c) \vee meet(a,c) \vee equal(a,c) \vee \\ coveredby(a,c) \vee cover(a,c) \vee overlap(a,c) &\leftarrow meet(a,b) \wedge meet(b,c). \end{aligned} \tag{2}$$

$$coveredby(a,c) \vee inside(a,c) \vee overlaps(a,c) \leftarrow meet(a,b) \wedge inside(b,c). \tag{3}$$

$$contains(a,c) \vee cover(a,c) \vee overlap(a,c) \leftarrow contains(a,b) \wedge coveredby(b,c). \tag{4}$$

$$inside(a,c) \vee coveredby(a,c) \leftarrow coveredby(a,b) \wedge coveredby(b,c) \tag{5}$$

$$contains(a,c) \vee covers(a,c) \leftarrow covers(a,b) \wedge covers(b,c). \tag{6}$$

From the examples above it is clear that the composition of two topological relationships can result in indefinite database clauses, i.e. a clause whose consequent (or conclusion) is the disjunction of more than one atom.

Our study is concerned only with definite deductive databases, and as such a valid clause is a clause with only one conclusion. Compositions of the above mentioned topological relationships can be transformed to the required form by noting the following observations, and applying the required transformations.

- The composition results in only one topological relationship. This case is directly transformed to a normal clause, and no transformation is needed. Clause 1 is an example of such a relation.
- The composition results in a subset of probable relationships, in which case one can deduce the negation of the improbable one(s), i.e. the impossibility of existence of the rest of the relationships set. For example, clause 2 can be rewritten as follows,

  $$\neg inside(a,c) \wedge \neg contains(a,c) \leftarrow meet(a,b) \wedge meet(b,c).$$

- The topological relationships *cover, coveredby, contains, inside and overlap* can be regarded as specializations of a relationship **go**verlap (general overlap), which indicates an intersection between the interior of the two point sets. Compositions resulting in the disjunction from this category viz., *cover* $\vee$ *contains* $\vee$ *overlap* and *coveredby* $\vee$ *inside* $\vee$ *overlap* can be generalized to a single relation goverlap. For example, clause 3 can be rewritten as follows,

  $$goverlap(a,c) \leftarrow meet(a,b) \wedge inside(b,c).$$

- Similarly the relationships *coveredby* and *inside* can be regarded as specializations of a relationship **gi**nside (general inside), which indicates that the interior of one set is a proper subset of the other. Compositions resulting in disjunctions of the form coveredby $\vee$ inside can be generalized to a single relationship ginside.
  For example, clause 5 can be rewritten as follows,

  $$ginside(a,c) \leftarrow coveredby(a,b) \wedge coveredby(b,c).$$

  The converse relationships *covers* and *contains* can be generalized to the same relationship *ginside* with the arguments interchanged, for example, clause 6 can be rewritten as follows,

  $$ginside(c,a) \leftarrow covers(a,b) \wedge covers(b,c).$$

- For the conjunction $< R_1(a,b) and R_2(b,c) >$ where no base relation is excluded, no definite database clause can be defined, for example,

$$\begin{aligned} disjoint(a,c) \vee meet(a,c) \vee equal(a,c) \ \vee \\ inside(a,c) \vee coveredby(a,c) \vee contains(a,c) \ \vee \\ covers(a,c) \vee overlap(a,c) \leftarrow disjoint(a,b) \wedge disjoint(b,c). \end{aligned}$$

Using the above observations, a transformation of the results in [8] is presented in the transitivity table shown in figure 4, which shows an $n \times n$ relationship composition matrix M. For example, $M_{3,2}$ (disjoint(a,c)) is the result of the composition of $M_{3,0}$ (inside(a,b)) and $M_{0,2}$ (meet(b,c)) and so on.

Deduction of negative relations cannot be expressed readily as *Horn clauses.* Deduction of unique positive atoms is probably the most useful in a GDB, and is the subject of our current research.

Consider an example database where *landuse, vegetation, rainfall,* and *slope* data layers are considered. If region #1 in a landuse data layer contains regions #55, covers region #25, and overlaps region #33 from the vegetation layer, then this can be expressed by the following set of clauses.

$\vdots$

$contains(landuse(\#1, urban), vegetation(\#55, grass)).$
$covers(landuse(\#1, urban), vegetation(\#25, none)).$
$overlap(landuse(\#1, urban), vegetation(\#33, wheat)).$

$\vdots$

$coveredby(vegetation(\#33, wheat), rainfall(\#101, 35)).$
$contains(vegetation(\#25, none), slope(\#1001, 5)).$
$meet(vegetation(\#55, grass), vegetation(\#57, grass)).$

$\vdots$

$goverlap(a,c) \leftarrow contains(a,b) \wedge meet(b,c).$
$contain(a,c) \leftarrow covers(a,b) \wedge contains(b,c).$
$goverlap(a,c) \leftarrow overlap(a,b) \wedge coveredby(b,c).$

One can deduce the following relations between the landuse layer and the rainfall, slope and vegetation layers,

$goverlap(landuse(\#1, urban), rainfall(\#101, 35)).$
$contains(landuse(\#1, urban), slope(\#1001, 5)).$
$goverlap(landuse(\#1, urban), vegetation(\#57, grass)).$

Thus deduction of useful spatial relationships can be done automatically without the need for the application of computational geometry algorithms.

(B) Randell's Logic Theory: Alternative approaches for spatial relationship representation can be used. For example, in [30] Randell et al introduce a theory based on a first order formalism for reasoning over space, time and processes. It

	d(b,c)	m(b,c)	i(b,c)	cb(b,c)	ct(b,c)	cv(b,c)	o(b,c)
d(a,b)	noinfo	¬ ct(a,c) ∧ ¬ cv(a,c)	¬ ct(a,c) ∧ ¬ cv(a,c)	¬ ct(a,c) ∧ ¬ cv(a,c)	d(a,c)	d(a,c)	¬ ct(a,c) ∧ ¬ cv(a,c)
m(a,b)	¬i(a,c) ∧ ¬cb(a,c)	¬i(a,c) ∧ ¬ct(a,c)	go(a,c)	¬ct(a,c) ∧ ¬cv(a,c)	d(a,c)	¬go(a,c)	¬ct(a,c) ∧ ¬cv(a,c)
i(a,b)	d(a,c)	d(a,c)	i(a,c)	i(a,c)	noinfo	¬ct(a,c) ∧ ¬cv(a,c)	¬ct(a,c) ∧ ¬cv(a,c)
cb(a,b)	d(a,c)	¬go(a,c)	i(a,c)	gi(a,c)	¬i(a,c) ∧ ¬cb(a,c)	¬i(a,c) ∧ ¬ct(a,c)	¬ct(a,c) ∧ ¬cv(a,c)
ct(a,b)	¬i(a,c) ∧ ¬cb(a,c)	go(a,c)	¬d(a,c)	go(a,c)	ct(a,c)	ct(a,c)	¬ct(a,c) ∧ ¬cv(a,c)
cv(a,b)	¬i(a,c) ∧ ¬cb(a,c)	¬ct(a,c) ∧ ¬cv(a,c)	go(a,c)	go(a,c)	ct(a,c)	gi(a,c)	go(a,c)
o(a,b)	¬i(a,c) ∧ ¬cb(a,c)	¬i(a,c) ∧ ¬cb(a,c)	go(a,c)	go(a,c)	¬i(a,c) ∧ ¬cb(a,c)	¬i(a,c) ∧ ¬cb(a,c)	noinfo

Fig. 4 Transitivity table for the set of base relations in figure 3, showing the transformation of composition results to one relationship or the conjunction of more than one relationship.

is the spatial part of this theory which is of relevance here. Ontological primitives include regions where every region coincides with a set of incident points, and is contained in a distinguished region called the universe. Unlike Egenhofer [8], the basic part of the formalism assumes one primitive dyadic relation C(x,y) read as 'x connects with y' which includes relationships between objects from external contact to identity in terms of mutually shared parts (this includes all the relationships in figure 3 except for the disjoint case) from which a basic set of dyadic relations are defined.

Some examples of this set expressed in IDB clauses would be,

$$\begin{aligned}
DisConnected(x,y) &\leftarrow \neg Connected(x,y).\\
ProperPart(x,y) &\leftarrow Part(x,y) \wedge \neg Part(y,x).\\
Identical(x,y) &\leftarrow Part(x,y) \wedge Part(y,x).\\
Overlap(x,y) &\leftarrow Part(z,x) \wedge Part(z,y).\\
ExternallyConnected(x,y) &\leftarrow Connected(x,y) \wedge \neg Overlap(x,y).
\end{aligned}$$

In terms of points incident in regions, C(x,y) holds when two regions connect; of the incident points contained in both regions, at least one incident point is shared. Compositions of topological relationships using the above definitions can be axiomatized in a similar manner.

In the representation formalism of [8], topological relationships between two point-sets, A and B, was described by nine possible set intersections (3 x 3 matrix) of A's boundary, interior, and complement with the boundary, interior, and complement of B. In order to establish a fact based on such relations using Egenhofer's model, a function is needed for describing the boundary, interior and the complement of an object in the geographic space. However, the problem is more complex in a GDB, where objects can be defined as specializations of point-sets (regions with holes) or as sets of other objects. In this case proving the composition of two topological relationships would be difficult, and consequently the composition matrices are more difficult to formulate. Note, however, that in a geographic space the 9-intersection matrix of [8] can be reduced to a 4-case intersection matrix by eliminating the complement intersections of the point-sets. Removing the complements in this case would neither affect the definition of the relationships nor their compositions, and thus greatly reduces the computation needed.

The power of Randell's formalism [30] can readily be recognized. If the connectivity relationship can be computed for the whole space, then a systematic derivation of the whole set of specialized relationships can be achieved without the need for the application of computational geometry algorithms, based solely on the satisfaction of the axioms defined. One can envisage a GDB where indexing structures can be used for the computation of the connectivity relationships for the space required, rules derive topological relationships between objects, and finally rules derive compositions of topological relationships as required.

Non-areal Spatial Relationships. Whatever formalism is used for defining spatial relationships, the main point to emphasize is that deduction mechanisms in a database can prove to be of major importance in the realization of large spatial databases in general and GDBs in particular. Although spatial relationships in the research work surveyed cover a basic representation primitive (a region), in a geographic context however, the line and point primitives possess the same functional and representational importance. Based on either formalism, a detailed study of topological relationships that these objects exhibit with themselves and the interrelationships between all the primitives is still needed.

For example, a line primitive has as a boundary its two end points and as interior the connection curve between its boundaries. Relationships may be defined on the basis of the boundary (represented by the two end points, ∂_1 and ∂_2) and the interior, denoted by $^\circ$ [2]. The intersection matrix in this case is,

$$I_n(A,B) = \begin{bmatrix} \partial_1 A \bigcap \partial_1 B & \partial_1 A \bigcap \partial_2 B & \partial_1 A \bigcap B^\circ \\ \partial_2 A \bigcap \partial_1 B & \partial_2 A \bigcap \partial_2 B & \partial_2 A \bigcap B^\circ \\ A^\circ \bigcap \partial_1 B & A^\circ \bigcap \partial_2 B & A^\circ \bigcap B^\circ \end{bmatrix}$$

Figure 5 shows some of the possible relationships. All the possible base relationships between 2 line primitives can be derived, and consequently transitivity tables formed. The same methodology can be applied to deriving relationships between different primitives, i.e. line and region, point and line and point and region. Of these the useful relationships in a geographic context can be extracted and represented by rules in the *IDB*.

Subjective Spatial Relationships. In a deductive database, the inexactness of the spatial relationships described in [28] resulting from the variety of

---

[2] notations as used in Egenhofer [11]

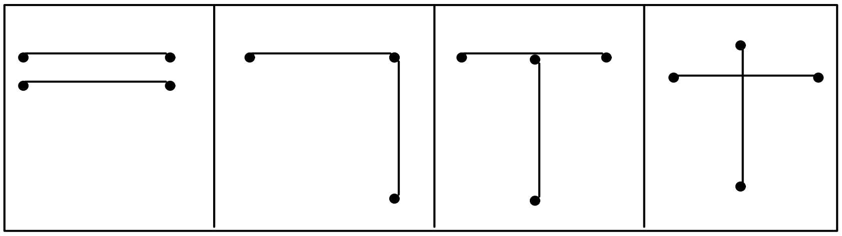

Fig. 5. Examples of relationships between two line primitives in 2-dimensional space

shape representations of spatial objects, (for example, in calculating the distance between two polygonal features, the problem is how to determine the points on the shape with which to apply the computation; is it the minimum, the maximum or the centroid distance that is required), can be resolved by defining rules in the database which correspond to different user views of the relationship. For example, in computing the distance between two cities, the definition of the relationships will differ if an areal representation is considered as opposed to a point representation. Also in the case of an areal representation, besides having to choose the point on each city object on which to carry out the computation, the definition would differ if the required distance is the shortest distance between the two points or the distance of the roads joining the cities, and so on.

Furthermore, subjective relationships which vary according to the class of objects considered can be expressed using rules.

$$\begin{aligned} near(building1, building2) &\leftarrow buffer\_zone(building1, 20, x) \\ &\wedge\ inside(building2, x) \\ near(city1, city2) &\leftarrow centroid\_distance(city1, city2, 100). \end{aligned}$$

Note that in a deductive object-oriented database, such relationships would be defined as methods on the appropriate object classes where inheritance of structure and behaviour is used.

3.2 Theme: Representing Different Themes of Interest

Information collected about geographic objects may be of different types. This is reflected in the different kinds of maps produced and used. Data sets representing the different types of information are referred to as *data layers*, and maps representing those data sets as *map layers*. Each data set might be analyzed and/or mapped individually. Alternatively, data sets may be combined to produce more meaningful information. For example, crop boundaries and types of soils can be used to determine the most productive soil for a particular crop. However, combining data sets can lead to an inordinate number of possible combinations.

Thus it is not practical to derive and represent instances of the results of such combinations of data sets apriori, as they could extend over the whole database space extension and might only be needed occasionally, if at all. Representation of object classes defined through the combination of more than one data set can be achieved through the use of rules, for example [2, 26],

$$\begin{aligned} cottage\_area(x) &\leftarrow forest\_vegetation(x) \wedge drained\_soil(x). \\ landslide\_susceptible\_area(x, high) &\leftarrow distance(x, active\_fault, 0..2) \\ &\wedge\ slope\_angle(x, v) \wedge v > 40 \\ &\wedge\ land\_cover(x, sparse\_vegetation) \\ &\wedge\ relative\_relief(x, w) \wedge w > 800 \\ &\wedge\ distance(x, ridge\_top, 0..900). \end{aligned}$$

Thus in non-geographic applications, object class definition is usually achieved through a pre-conceived template of properties and relationships where instances of the object class are explicitly created and attached to a class. For example, *X is an instance of person, and Y is an instance of a car.* In the geographical domain, however, instances of a geographic object class would have to be inferred according to the rules, rather than simply stated. Overlay operations and rules for deriving relationships are used to find instances of such object classes. The resulting objects are generally physically colocated with part-of or all-of other objects in the database, i.e. they have arbitrary spatial shapes depending on their definition and constraints. The deductive definition of thematic concepts eases experimentation with alternative criteria, which increases the number of ways in which the underlying data can be used.

3.3 Resolution: Representing Different Details (or Scales) of Interest

For practical reasons, most maps are scaled representations. It is impossible to represent all geographic information on a one-to-one basis, so a scale is devised to retain the data required and to present it as a map of a particular size. The amount of detail required in the representation determines the choice of a particular representation scale. Too many irrelevant details can hinder the conveyance of the information in the map. For example, a map representation which shows the unemployment distribution, which is usually collected for large areas, need not show the name of every street in the area.

The process of scaling the map can involve screening out some details from a spatial scene as well as using different spatial representations of objects, for example, merging shapes or transforming an area to a line or a point, etc. The resolution dimension can be seen from two points of view,

- Cartographic: scaled representation of database objects used solely for cartographic purposes.
- Analytic: where a different spatial representation of the geographic object is needed to model the application under analysis, for example, studying a problem of path finding using a graph to represent the road network, and planning maintenance schedules using an area representation of the road network.

Although small scale representations of geographic objects can be derived from high resolution representations using generalization rules [22], the storage of multiple representations of geographic objects is sometimes necessary where situations involving large degrees of generalization can cause delays in computation which could not be tolerated in an interactive GIS. Representing multiple representations of objects in a GDB is a problem which is not readily handled using existing data models and is part of our research goals in this project, where extensions are sought to the object data model presented in [13].

In the context where an object can have more than one spatial representation several problems arise, which include maintaining correct links between related multi-represented objects, maintaining database integrity during update, and choosing the appropriate representation during manipulation. Integrity constraint-type rules can be formulated to handle the first two cases, while specific database rules are used for the third. For example,

$$\begin{aligned} spatial(X, Scale, Rep) &\leftarrow equal(Scale, 1:50,000) \\ &\wedge\ point(X, Rep). \\ spatial(X, Scale, Rep) &\leftarrow equal(Scale, 1:1250) \\ &\wedge\ region(X, Rep). \\ route(X, Y, S) &\leftarrow scale(S) \\ &\wedge\ spatial(X, S, P) \wedge spatial(Y, S, R) \\ &\wedge\ path(P, R). \end{aligned}$$

where different object representations and different levels of detail are associated to each representation scale.

3.4 Time: Representing the Change in the Data Over Time

Maps are also used to reflect the change in features over time whether as a record of past events (historical maps) or as prediction of future events (modeling and planning maps). Modeling the change of the data over time is a general requirement in any database system. A study of the implication of this dimension in a geographic database is not considered in the scope of our research.

4 Deduction in GDB Queries

Although conventional query languages such as SQL have been successfully used as query languages for many applications which can be easily expressed in terms of tables, its use is very limited when dealing with new applications such as image databases and GISs which need more complex underlying structures than tables. Extensions to such query languages have been proposed to cope with properties of spatial data [16, 5]. Such extensions are considered unnatural and at best short term solutions, as the real problem lies in forcing spatial concepts into a framework designed for data modelled as tables. Using a suitable geographic data model such as an object-oriented model coupled with deductive capabilities, as proposed here, provides an effective framework for the expression of queries against geographic data.

Three major advantages can be observed from the use of a logic query language for expressing geographic queries:

Firstly, declarative expression of geographic queries is offered using a first order language. An example of a typical GDB query is to find the objects satisfying particular spatial and aspatial conditions. For example, to select potential areas for waste disposal sites the following query can be used:

$$\begin{aligned} \leftarrow \quad & close(x, waste\_source) \wedge close(x, railway\_station) \\ & \wedge\, close(x, main\_road) \wedge inside(x, low\_quality\_agricultural\_buffer) \\ & \wedge\, distance(x, residential\_dwellings, y) \wedge y >= 500 \\ & \wedge\, \neg site\_special\_scientific\_interest(x). \end{aligned}$$

As shown in the above query, the evaluation of a spatial relationship can be invoked directly (distance/3) or from within other clauses (close/2). In both cases the spatial relationship is implemented using computational geometry. Such expressions are possible in our database system [12] where two languages based on the same data model coexist, viz. a logic language used for logical expression of queries and an imperative database programming language for implementing methods (including spatial) over geographic objects. The logic language can invoke methods defined in the imperative language and logic language expressions can be embedded in the imperative language.

Secondly, recursive queries which cannot be expressed using conventional query languages are directly expressible using the logic language. This type of query is essential for expressing network-oriented queries which are a subset of GIS queries which can be modeled using *Horn clauses*. For example, a query such as "What are the common parts of the paths between London and Edinburgh using British Rail *(BR)*" can be evaluated by finding all *BR* paths (recursive definition) between London and Edinburgh and then recursively selecting all the common path segments from the resulting path list.

Finally, expressing derived data in the *IDB* using the same language as that which is used for querying provides two major advantages, viz.

- Queries against geographic data can directly call rules in the *IDB*, which results in easier query formulation.
- Queries formulated during geographic analysis can be stored as derived data in the *IDB*, and can thus be reused for future analysis and manipulation.

Both features are unique to a deductive database and are useful in a GDB where complex spatial queries and results of queries can be used to enrich the database.

5 From the Primitive to the Object level: A Deductive Approach

As noted in section 2, data input to a GDB may come from different sources, whether as "*paper maps*" or "*images*". In either form, the data has to pass through a process of digitization to produce a result in computer-readable form. The result of this digitization process is a geographic data set at the primitive level. An interpretation process is needed to transform the geographic data from this level to the object level needed for a GDB, as shown in figure 6. This task can be aided by automatic object recognition, which is important for the following reasons:

1. Currently the capturing and updating of data for a GDB is based on multiple primitive data sources. The data is initially stored at the primitive level, and later during an update process, changes to the primitive level must be propagated manually to the corresponding object level objects.
2. The huge amount of data stored and the frequency of its update makes a manual interpretation process costly and limited in its application.
3. The existence of data at the object level is essential in a GIS environment where one of the requirements is to have a user interface and a query language which are insensitive to the underlying geometrical representation of objects (i.e. the primitive level of representation).

The above facts support the inclusion of an object recognition module as a component of a GDB system. This module is essentially deductive in nature, where the hypotheses used by the map reader are mapped naturally into rules for automatic recognition. Rules in this object recognition module require access to computational geometry algorithms defined on the objects (primitive level) in the *EDB* during compilation.

Using deductive rules for object extraction from the primitive data representation means:

1. Detaching the recognition criteria from low-level algorithms.
2. Expressing the extraction criteria more concisely.
3. Easier modification of extraction criteria.

A rule-based approach for automatically extracting road networks from Ordnance Survey ‘OS’ large scale (1:1250) maps has been implemented in [1] and is used as the preprocessing stage for data input to the GDB.

Approaches to object (feature) recognition from multiple data sources, both *paper maps* and *images*, that have been investigated in the literature can be classified according to the properties utilized, into: *attribute-based map interpretation* and *structure-based map interpretation*. Both are discussed below.

5.1 Attribute-Based Object Recognition

This approach is suitable for object classes which can be described by a set of features or measurable attributes in isolation from other object classes. The power of this approach depends upon:

1. The availability of attributes for the object type that are invariant. For example, for any type of road, approximate parallelism of its sides is always a valid property.

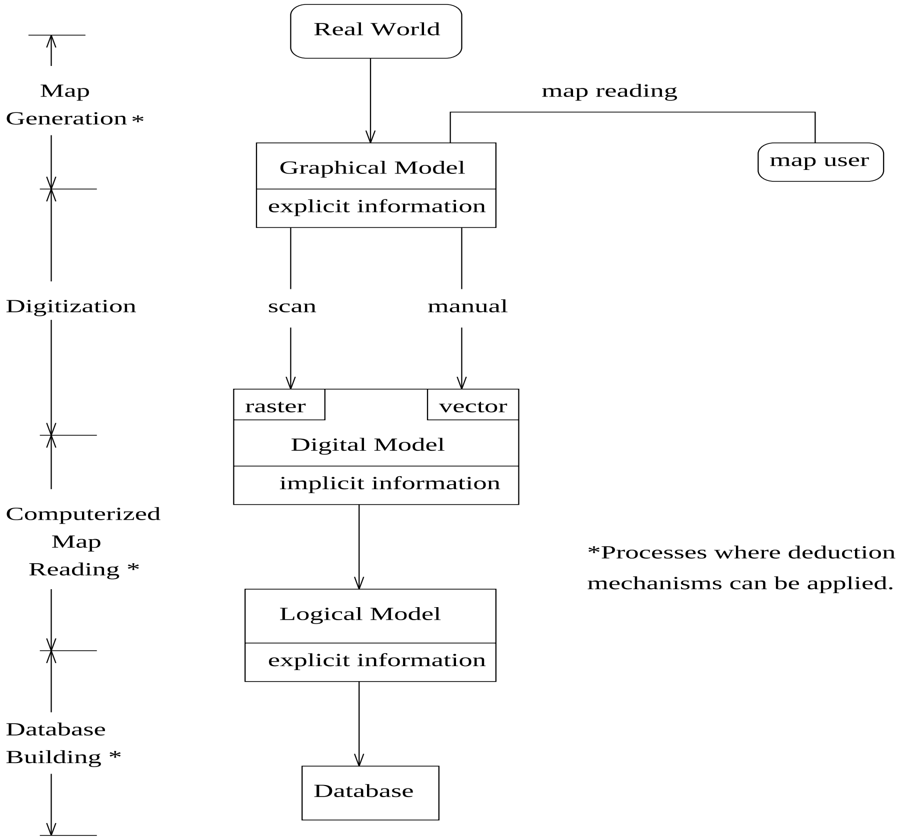

Fig. 6. The process of converting a map of the real world into a database representation

2. The amount of discriminatory information contained in the properties. The basic recognition process usually results in noise objects which are incorrectly identified as instances of the original object class. This is due to the fact that not all the characteristics of an object class are utilized in the interpretation process, as this may result in redundancy and inefficiency. More attributes can then be used to eliminate the noise objects.

The first step in this approach is to have a suitable classification and careful choice of the object attributes needed for the recognition process. If a complete set of discriminatory attributes for each object class can be determined from the data, then the recognition and classification of objects may be reduced to a matching process which is essentially a "table look-up" scheme. Since each pattern of object classes is considered in isolation from other patterns, there is no use for a specific search strategy as the order of extraction does not affect the interpretation process, and the matching is done for every shape of interest. An obvious disadvantage of this method is that it is a single step method where extracted information is not utilized for other object interpretation.

One classification of spatial attributes of an object is between *qualitative* and *quantitative*. Qualitative or descriptive properties, such as parallelism or curvature, smoothness, and homogeneity are obvious to a human interpreter, but can be computationally time consuming to determine from a digital model as they involve evaluation of the property on the micro level for the whole object. On the other hand, the quantitative or objective properties, such as the area of a polygon, length of a line, etc., are computationally easier to evaluate, although they are probably not as obvious to a human interpreter as qualitative properties. In most cases it is the qualitative properties which most surely distinguish an object type, which means that high computation overheads will be a particular characteristic of this approach.

Examples of extraction criteria utilizing only the attributes of geographic phenomena can be found in the [3, 24, 17, 21, 34]. An example of criteria used in [4] can be expressed as rules as follows,

A shape is a railway if
It has a uniform width equal to the standard gauge of railways, and
Sides have certain degree of straightness, and
Sides are parallel, and
It has no more than 4 sharp turns .

A shape is a pavement if
Sides are parallel, and
Each side is almost straight, and
Has a narrow width within a certain limit.

A shape is a house if
Its width is of the same order of magnitude as its length, and
Its area is within a certain range, and
The ratio of the area to the perimeter is within a certain range.

The disadvantages of relying solely on an attribute-based approach can be summarized as follows,

1. It fails to extract complex map objects, as they are usually characterized by complex spatial structure. The recognition of such objects necessitates the extraction of spatial relationships between objects.
2. It can be ineffective if object classes with no strong discriminative attributes are extracted. For example, objects that can have similar shapes (a garage, or a garden as big as a house).
3. It can become inefficient when attributes are expensive to compute, as it involves repeating the same evaluation process for every existing shape.

5.2 Structure-Based Object Recognition

As described earlier, spatial structure is the pattern of existence of objects in specific spatial relationships with respect to each other. This can be attributed to the fact that some real world objects are functionally related (for example, houses having access to roads, and bridges connect roads), while others are naturally structured (such as drainage patterns). The structure-based approach to map interpretation involves the utilization of both spatial attributes of objects and spatial relationships between objects. This ensures recursive use of properties for extracting new objects on a map. This approach normally involves the application of a search strategy, where a search strategy can be defined as the process of selecting objects and defining the order in which the interpretation process is carried out.

Two different strategies can be recognized,

1. Identify an object class and then use spatial relationships between objects to extract instances of related object classes and so on, which is the approach followed by [17, 32, 19].
2. Identify a distinctive complex structure and then divide it up into its subparts using knowledge of their properties, which is the approach followed by [18].

The choice of a particular search strategy in structure-based interpretation is crucial to the efficiency and sometimes the success of the interpretation process.

The second strategy has been implemented [1] to extract the road network from large scale maps where the network is extracted as a complete object using spatial relationships between land parcels and the road boundaries. Then the first strategy is used to divide the network into its individual roads for the assignment of postal addresses. To achieve this, houses are grouped in chains to indicate that they should have the same road name in the address. Chains are classified according to their shapes, for example, closed chains indicate houses in a cul-de-sac. Thus,

$$\begin{aligned} enclosed(RoadSeg, Chain) \leftarrow\ & closed\_loop\_chain(Chain) \wedge \\ & junction(J, RoadSeg) \wedge \\ & point\_in\_polygon(J, Chain). \end{aligned}$$

The geometrical relationships between road segments are used to extend road names when the above method fails using the following rules,

$$\begin{aligned} extend\_name(S_1, S_2) &\leftarrow dead\_end\_extension(S_1, S_2). \\ extend\_name(S_1, S_2) &\leftarrow same\_alignment(S_1, S_2). \end{aligned}$$

6 Conclusion

In this paper, the application of deduction to a GDB has been presented. This was based on a comprehensive analysis of the requirements for a GDB in terms of representation and data input. Four dimensions which need to be catered for within a GDB have been identified, namely, space, theme, resolution and time. Deduction in a GDB was found to be useful for:

1. Defining a complex geographic phenomenon through its spatial structure, i.e. the spatial relationships between its component objects.
2. Reasoning over the geographic space for the inference of implicit spatial relationships which otherwise are not generally defined over object classes and would require the application of computational geometry algorithms.
3. Defining geographic object classes which extend over more than one data layer, thus obviating the need to explicitly create instances of such numerous object combinations.
4. Defining generalization rules for extracting one spatial object representation from another for cartographic and analytical purposes.
5. Declarative and recursive formulation of GDB queries, for storing frequently asked queries as rules in the database.
6. Automatic extraction of object level concepts from primitive level representations, which is a necessary operation in a GDB either for initial data loading or subsequent map updating.

In the DOOD project, the integration of deductive and object-oriented approaches in the design of spatial databases is proposed. Our intention is to demonstrate the usefulness of deduction for spatial databases (in particular GDBs) focusing on issues 1,2,3,5 and 6 above. Towards this aim, a prototype rule-based object extraction module for road network extraction and naming has been implemented. In addition a hypothetical resource management and allocation geographic application has been designed and is currently being implemented to test the GDB.

Acknowledgments The work that resulted in this paper has been funded by the Science and Engineering Research Council through the IEATP programme, and their support is duly acknowledged. Many improvements were suggested by our colleagues A.A.A. Fernandes and M.L. Barja, for which we are sincerely thankful. We would also like to thank Mr Neil Smith of Ordnance Survey and Dr. J.M.P. Quinn representing ICL, as the industrial partners in the project, for their assistance and support and useful discussions on the subject of this paper. Finally, thanks are due to Dr. Keith G. Jeffery of Rutherford Appleton Laboratory for his support and advice on the subject.

References

1. A.I. Abdelmoty, M.H. Williams, and J.M.P Quinn. A Rule-Based Approach To Computerized Map Reading. to be published in Information and Software Technology, 1993.
2. S. Arnoff. *Geographic Information Systems: A Management Perspective.* WDL Publications, Ottawa, Canada, 1989.
3. M. David and Jr. McKeown. Towards Automatic Cartographic Feature Extraction. In L.F. Pau, editor, *Mapping and Spatial Modelling for Navigation*, pages 150–180. Springer-Verlag, 1990.
4. M. De Simone. *Data Structures and Feature Recognition: From the Graphic Map to a Digital Database.* PhD thesis, North East London Polytechnic, 1985.

5. M.J. Egenhofer. An Extended SQL Syntax to Treat Spatial Objects. In *Second International Seminar on Trends and Concerns of Spatial Sciences*, pages 83–95, 1987.
6. M.J. Egenhofer. A Formal Definition of Binary Topological Relationships. In *Proc. 3rd International Conference on Foundations of Data Organization and Algorithms, FODO 89*, Paris, June 1989.
7. M.J. Egenhofer. Deficiencies of SQL as a GIS Query Language. In D.M. Mark and A.U. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space*, pages 477–491. NATO ASI Series, 1990.
8. M.J. Egenhofer. Reasoning About Binary Topological Relations. In O. Gunther and H.J. Scheck, editors, *Advances in Spatial Databases, 2nd Symposium, SSD'91*, Lecture Notes in Computer Science, 525, pages 143–161, Zurich, Switzerland., 1991. Springer-Verlag.
9. M.J. Egenhofer and A. Frank. Object-Oriented Databases: Database Requirements for GIS. In *Proc. Int. GIS Symposium: The Research Agenda*, volume 2, pages 189–211, 1987.
10. M.J. Egenhofer and R.D. Franzosa. PointSet Topological Spatial Relations. *Int. J. Geographic Information Systems*, 5(2):161–174, 1991.
11. M.J. Egenhofer and J.R. Herring. A Mathematical Framework for the Definition of Topological Relationships. In *Proceedings of the 4th international Symposium on Spatial Data Handling*, volume 2, pages 803–13, 1990.
12. Alvaro A.A. Fernandes, Maria L. Barja, Norman W. Paton, and M. Howard Williams. An Object-Oriented Database for Large-Scale Application Development, 1993. To appear in *Proceedings of the Eleventh British National Conference on Databases (BNCOD 11)*, LNCS, Springer-Verlag.
13. Alvaro A.A. Fernandes, M. Howard Williams, and Norman W. Paton. A Formal Abstract Definition of Objects as a Data Modelling Primitive. Technical Report TR92003, Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland, April 1992. Revised June 1992.
14. A.U. Frank. Spatial Concepts, Geometric Data Models and Data Structures. Technical Report 90-11, National Center Geographic Information and Analysis, 1990. Two Perspectives on Geographical Data Modelling.
15. R.H. Guting. Gral: An Extensible Relational Database System for Geometric Applications. In *Proceedings of the Fifteenth International Conference on Very Large Data Bases*, pages 33–44, 1989.
16. J. Herring, R. Larsen, and J. Shivakumar. Extensions to the SQL Language to Support Spatial Analysis in a Topological Database. In *GIS/LIS'89*, pages 741–750, 1989.
17. L.M. Jensen. Knowledge-Based Classification of an Urban Area Using Texture and Context Information in Landsat-TM Imagery. *Photogrammetric Engineering and Remote Sensing*, 56(6):899–904, 1990.
18. D.M. Mckeown, W.A. Harvey, and J. Mckdermot. Rule-Based Interpretation of Aerial Imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7:570–585, 1985.
19. G. Mehldau and R. Schowengredt. A C-Extension for Rule-Based Image Classification Systems. *Photgrammetric Engineering and Remote Sensing*, 56(6):887–892, 1990.
20. L. Mohan. An Object Oriented Knowledge Representation for Spatial Information. *IEEE Transactions on Software Engineering*, 14:675–80, 1988.

21. O.A. Morean and R. Kasturi. Symbol Identification in Geographical Maps. In *Seventh International Conference on Pattern Recognition*, volume 2, pages 966–7, USA, 1984. SILVER SPRING.
22. J.C. Muller. Rule Based Generalization: Potentials and Impediments. In *Proceedings of the Fourth International Symposium on Spatial Data Handling*, volume 1, pages 317–334. IGU, 1990.
23. J. Munkers. *Elementary Differential Topology.* Princeton University Press, Princeton, NJ, 1966.
24. T. Nagao, T. Agui, and M. Nakajima. Automatic Extraction of Roads Denoted by Parallel Lines from 1/25,000-Scaled Maps Utilizing Skip-Scan Method. *SYST. COMPUT. JPN. (USA)*, 21(11):96–105, 1990.
25. B.C. Ooi. Efficient Query Processing in Geographic Information Systems. In G. Goos and J. Hartmanis, editors, *Lecture Notes in Computer Science, 471.* Springer Verlag, 1990.
26. A.K. Pachauri and P. Manoj. Landslide Hazard Mapping based on Geological Attributes. *Engineering Geology*, 32:81–100, 1992.
27. D.J. Peuquet. A Conceptual Framework and Comparison of Spatial Data Models. *Cartographica*, 21(4):66–113, 1984.
28. D.J. Peuquet. The Use of Spatial Relationships to Aid Spatial Database Retrieval. In *Proc. 2nd Int. Symp. on Spatial Data Handling*, pages 459–71, 1986.
29. D.V. Pullar and M.J. Egenhofer. Towards Formal Definition of Topological Relations Among Spatial Objects. In *Proc. 3RD International Symposium on Spatial Data Handling*, pages 225–241, Sydney, Australia, 1988.
30. D. A. Randell, Z. Cui, and Cohn G. A Spatial Logic Based on Regions and Connection. In *Proccedings of the third International Conference on Principles of Knowledge Representation and Reasoning KR'92.* 1992. Yet to appear.
31. H. Samet. Heirarchial Data Structures For Spatial Reasoning. In *Mapping and Spatial Modeling for Navigation*, pages 41–58. Springer-Verlag, 1990.
32. T. Schenk and O. Zierstein. Experiments with a Rule-Based System for Interpreting Linear Map Features. *Photogrammetric Engineering and Remote Sensing*, 56(6):911–917, June 1990.
33. T.R. Smith, R. Ramakrishnan, and A. Voisard. Object-Based Data Model and Deductive Language for Spatio-Temporal Database Applications. In G. Gambosi, M. Scholl, and H.W. Six, editors, *Database Management Systems for Geographical Applications*, 1991.
34. S. Suzuli and T. Yamada. MARIS: Map Recognition Input System. In L.F. Pau, editor, *Mapping and Spatial Modelling for Navigation*, pages 95–116. Springer Verlag, 1990.
35. M.F. Worboys, H.M. Hearnshaw, and D.J. Maguire. Object-Oriented Data and Query Modeling for Geographical Information Systems. In *Proceedings of the Fourth International Symposium on Spatial Data Handling*, pages 679–688, Zurich, 1990.