A Small Set of Formal Topological Relationships Suitable for End-User Interaction

Eliseo Clementini[1], Paolino Di Felice[1], and Peter van Oosterom[2]

[1] Università di L'Aquila, Dipartimento di Ingegneria Elettrica,
67040 Poggio di Roio, L'Aquila, Italy
Email: clementini@vaxaq.cc.univaq.it
pdifelice@vxscaq.aquila.infn.it

[2] TNO Physics and Electronics Laboratory,
P.O. Box 96864, 2509 JG The Hague, The Netherlands
Email: oosterom@fel.tno.nl

Abstract. Topological relationships between spatial objects represent important knowledge that users of geographic information systems expect to retrieve from a spatial database. A difficult task is to assign precise semantics to user queries involving concepts such as "crosses", "is inside", "is adjacent". In this paper, we present two methods for describing topological relationships. The first method is an extension of the geometric point-set approach by taking the dimension of the intersections into account. This results in a very large number of different topological relationships for point, line, and area features. In the second method, which aims to be more suitable for humans, we propose to group all possible cases into a few meaningful topological relationships and we discuss their exclusiveness and completeness with respect to the point-set approach.

1 Introduction

In the context of geographic information systems (GISs), the spatial relationships existing between the geographic objects play a central role both at the query definition level and at the query processing level. In fact, the easiest way for users to define spatial queries is based on the possibility of expressing spatial conditions among geographic objects (e.g., adjacency of regions) inside the query statement.

The need to refer to spatial relationships arises a second time when the database management system (DBMS) tries to process a spatial query. Obviously, spatial queries can be easily processed if all the geometric relationships between the objects of interest are explicitly stored; however, such a choice is unsatisfactory since it requires a tremendous amount of disk space and, furthermore, it implies the execution of time-consuming maintenance procedures. It follows that instead of storing all spatial relationships among the objects of interest it is more convenient to compute them. To that purpose, a deep understanding of how to evaluate spatial relationships is needed.

The need for developing a sound mathematical theory of spatial relationships to overcome the shortcomings of almost all geographic applications was clearly stated by Abler several years ago [Abl87]. Nevertheless, the exploration/formalization of spatial relationships is still an open problem, and a multi-disciplinary effort involving linguists and psychologists besides geographers and computer scientists is probably the best approach to get good results.

So far there is a good, but still incomplete, understanding of *topological relationships*, that is the subset of spatial relationships characterized by the property of being preserved under topological transformations, such as translation, rotation, and scaling. In the literature, we find several attempts to describe a set of meaningful topological relationships (see, among others [RFS88, MS89, B$^+$91, CDD91]), but it is difficult to find a formal definition of them. A good formal approach can be found in [EF91], that has been extended in [EH92], where the authors adopt a method to give exact semantics to the binary topological relationships based on the point-set theory. A drawback of this method is that they distinguish only between empty or non-empty intersections of boundaries and interiors of geometric objects, and also the method results in too many different relationships to be used by end-users. This will become even worse if the method of Egenhofer is extended in order to take into account the dimension of intersections. The list of cases that results from this approach is not directly related to the user interpretation of topological facts. In [ME92], after a testing experience with human subjects, the authors conclude that there is a significant connection between human interpretation of spatial relationships and the Egenhofer method. However, a way of grouping relationships is needed in order to map concepts from a geometric level to a higher (user-oriented) level.

In the present contribution, we take into account the dimension of the result of the intersection (*dimension extended method*); furthermore, our objective is to keep the resulting number of potential topological relationships as small as possible. To achieve the latter goal, we grouped together the relationships (that are somehow similar) into a few more general topological relationships: *touch*, *in*, *cross*, *overlap*, and *disjoint*. We called this approach the *calculus-based method*, since it uses the constructs of the Object-Calculus introduced in a previous paper [CD92]. The five relationships are overloaded concepts in the sense that they may be used for point, line, and area type of features. Further, more detailed distinctions among topological situations are possible by introducing operators on the *boundary* of features. Specifically, it is possible to use directly circular lines (coming from the boundaries of areas) and end-points (coming from the boundaries of lines).

The paper is structured as follows. Section 2 contains general definitions for the Object-Calculus and for the geometric point-set theory approach. Section 3 first recalls the original Egenhofer method and hence it discusses the dimension extended method. In Section 4, we give the exact semantics to the five basic topological relationships and several examples of usage of them; then we prove that the five relationships are mutually exclusive (e.g., it cannot be the case that two features are involved in an *in* and *overlap* relationship with each-other) and that there are no cases that fall outside them. Furthermore, we prove that a combination of these terms, together with a boundary operator for line and area features, is expressive enough to represent all possible cases in the dimension extended method. A brief description of the implementation is given in Section 5. Section 6 contains a discussion about the possible extensions.

2 General Definitions

The notations P, L, and A are used for point, line, and area features. If it is necessary to distinguish between two features of the same type, then numbers are used; e.g. A_1 and A_2. The symbol λ is used in situations where it may represent one of the three feature types.

In [CD92], we proposed the Object-Calculus, which is a formal query language suitable for querying geographical databases. In such a calculus, the notation $\langle \lambda_1, r, \lambda_2 \rangle$ means that the features λ_1 and λ_2 are involved in the relationship r; we call this triplet a *fact*. Facts can be combined through the *and* ($\wedge$), *or* ($\vee$) Boolean operators. Besides stating facts, the Object-Calculus allows the usage of methods (oprations) inside a query statement. Let m be a method and I a specific instance of a feature type λ, the pair (I, m) means that the method m operates on the instance I, and returns a new instance, say I_1. We overload the notation (I, m) to denote also the resulting instance I_1.

Formal definitions of geometric objects (features) and relationships are based on the point-set approach, where features are sets and points are elements of these sets (see [Kel55]) for a general topology reference). The subject of the relationships are the "simple" points, lines, and areas commonly used in GISs: the topological space is $\mathbb{R}^2$; all kinds of features are closed sets, that is, each feature contains all its accumulation points (also called limit points); also all features are connected, that is, they are not the union of two separated features. Specifically:

1. area features are only connected areas with no holes;
2. line features are lines with no self intersections and either circular (closed curves) or with only two end-points;
3. point features may contain only one point.

We consider a function "dim", which returns the dimension of a point-set. In case the point-set consists of multiple parts, then the highest dimension is returned. Note that this can only be the case for intermediate point-sets as our features always consist of one part. In the following definition, S is a general point-set, which may consist of several disconnected parts:

$$dim(S) = \begin{cases} - & \text{if } S = \emptyset \\ 0 & \text{if } S \text{ contains at least a point and no lines or areas} \\ 1 & \text{if } S \text{ contains at least a line and no areas} \\ 2 & \text{if } S \text{ contains at least an area} \end{cases} .$$

The boundary and the interior of features are used in the Egenhofer method for describing the topological relationships. The same is true for our approach; therefore, we give definitions of boundary and interior for the three types of features that are slightly different from the pure mathematical theory, but lead to consistent definitions for relationships. The boundary of a feature λ is denoted by $\partial\lambda$. It is defined for each of the feature types as follows:

1. ∂P: we consider the boundary of a point feature to be always empty;
2. ∂L: the boundary of a line is an empty set in the case of a circular line while otherwise is the set of the two separate end-points;
3. ∂A: the boundary of an area is a circular line consisting of all the accumulation points of the area.

The interior of a feature λ is denoted by λ°. It is defined as:

$$\lambda^\circ = \lambda - \partial\lambda \ .$$

Note that the interior of a point and of a circular line is equal to the feature itself.

3 The Dimension Extended Method

Egenhofer [EF91] originally described his method for classifying topological binary relationships between area features. The classification is based on the intersections of the boundaries and interiors of the two features. These are represented by the four sets:

$$\begin{aligned} S_1 &= \partial A_1 \cap \partial A_2 \\ S_2 &= \partial A_1 \cap A_2^\circ \\ S_3 &= A_1^\circ \cap \partial A_2 \\ S_4 &= A_1^\circ \cap A_2^\circ \ . \end{aligned}$$

Each of these four sets may be empty ($\emptyset$) or non-empty ($\neg\emptyset$). This results in a total of $2^4 = 16$ combinations (Table 1), which may not all result in a valid topological relationship, because of the properties of area features. As there are 8 impossible cases (proved in [EF91]) and 2 pairs of converse relationships, the number of different types of relationships is 6: disjoint, in, touch, equal, cover, and overlap. Figure 1 gives a pictorial representation of these six relationships. One of the good aspects of this approach is that it gives an exact definition of the mentioned relationships. Also, it takes into account all possible combinations of intersections (a form of completeness).

The first extension to the standard approach is to add also point and line features, resulting in 6 major groups of binary relationships: area/area (as described above), line/area, point/area, line/line, point/line, and point/point. This approach has been described in [dHvO92, EH92, ?]. A drawback of the approach is the large number of different relationships, of which each has its own name. As it may be hard to remember all these names, the users might become confused.

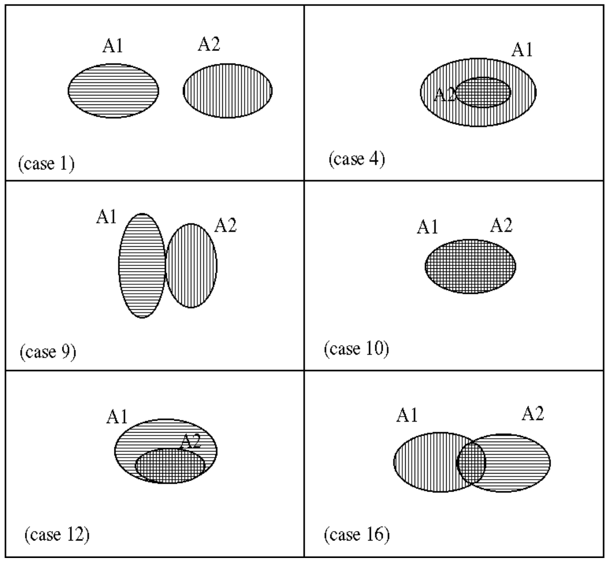

Fig. 1. A visualization of the six different relationships in Table 1

Table 1. The range of area/area situations as in the original Egenhofer method

case	S_1 $\partial A_1 \cap \partial A_2$	S_2 $\partial A_1 \cap A_2^\circ$	S_3 $A_1^\circ \cap \partial A_2$	S_4 $A_1^\circ \cap A_2^\circ$	relationship name
1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	A_1 disjoint A_2
2	$\emptyset$	$\emptyset$	$\emptyset$	$\neg\emptyset$	
3	$\emptyset$	$\emptyset$	$\neg\emptyset$	$\emptyset$	
4	$\emptyset$	$\emptyset$	$\neg\emptyset$	$\neg\emptyset$	A_2 in A_1
5	$\emptyset$	$\neg\emptyset$	$\emptyset$	$\emptyset$	
6	$\emptyset$	$\neg\emptyset$	$\emptyset$	$\neg\emptyset$	A_1 in A_2
7	$\emptyset$	$\neg\emptyset$	$\neg\emptyset$	$\emptyset$	
8	$\emptyset$	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	
9	$\neg\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	A_1 touch A_2
10	$\neg\emptyset$	$\emptyset$	$\emptyset$	$\neg\emptyset$	A_1 equal A_2
11	$\neg\emptyset$	$\emptyset$	$\neg\emptyset$	$\emptyset$	
12	$\neg\emptyset$	$\emptyset$	$\neg\emptyset$	$\neg\emptyset$	A_1 cover A_2
13	$\neg\emptyset$	$\neg\emptyset$	$\emptyset$	$\emptyset$	
14	$\neg\emptyset$	$\neg\emptyset$	$\emptyset$	$\neg\emptyset$	A_2 cover A_1
15	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	$\emptyset$	
16	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	$\neg\emptyset$	A_1 overlap A_2

Another drawback of this method is that it is impossible to distinguish between certain cases, which are usually regarded as different by users. For example, two areas that have one point in common, and two areas that have a complete line in common, do both fall under the same "touch" relationship, because the intersection of their boundaries (S_1) is non-empty and the other intersections (S_2, S_3, and S_4) are all empty (case 9 in Table 1).

Table 2. The line/area situations in the dimension extended method

case	S_1 $\partial A \cap \partial L$	S_2 $\partial A \cap L^\circ$	S_3 $A^\circ \cap \partial L$	S_4 $A^\circ \cap L^\circ$	possible
1	-	-	-	-	yes
2	-	-	-	1	no
3	-	-	0	-	no
4	-	-	0	1	yes
5	-	0	-	-	yes
6	-	0	-	1	yes
7	-	0	0	-	no
8	-	0	0	1	yes
9	-	1	-	-	yes
10	-	1	-	1	yes
11	-	1	0	-	no
12	-	1	0	1	yes
13	0	-	-	-	yes
14	0	-	-	1	yes
15	0	-	0	-	no
16	0	-	0	1	yes
17	0	0	-	-	yes
18	0	0	-	1	yes
19	0	0	0	-	no
20	0	0	0	1	yes
21	0	1	-	-	yes
22	0	1	-	1	yes
23	0	1	0	-	no
24	0	1	0	1	yes

In the dimension extended method, we take into account the dimension of the intersection, instead of only distinguishing empty or non-empty intersections. In order to illustrate this extension, the line/area type of topological relationships will be elaborated on. In two-dimensional space, the intersection set S can now be either $\emptyset$ (empty), 0D (point), 1D (line), or 2D (area). At first sight, these 4 possibilities might result into $4^4 = 256$ different cases. Fortunately, a lot of cases are impossible and only the following are possible:

$$\begin{array}{ll} S_1 = \partial A \cap \partial L : \emptyset, \text{or } 0D & (2 \text{ cases}) \\ S_2 = \partial A \cap L^\circ : \emptyset, 0D, \text{ or } 1D & (3 \text{ cases}) \\ S_3 = A^\circ \cap \partial L : \emptyset, \text{or } 0D & (2 \text{ cases}) \\ S_4 = A^\circ \cap L^\circ : \emptyset, \text{or } 1D & (2 \text{ cases}) \ . \end{array}$$

This is due to the fact that the dimension of the intersection cannot be higher than the lowest dimension of the two operands of the intersection; $dim(\partial A) = 1$, $dim(A^\circ) = 2$, $dim(\partial L) = 0$, and $dim(L^\circ) = 1$. Further, the definitions of line and area features exclude the option that $dim(S_4) = 0$. Therefore, instead of 256, there are only $2 * 3 * 2 * 2 = 24$ possible cases. Table 2 shows that only 17 out of these 24 cases are really possible.

Cases 3, 7, 11, 15, 19, and 23 are impossible because, if the intersection of the interior of an area with the boundary of a line (S_3) results in a point (0D), then it is impossible that the intersection of the interiors (S_4) is empty. Case 2 is impossible because if the intersection of the interiors (S_4) results in a line, then the other sets (S_1, S_2, and S_3) cannot all be empty. Note that in Table 2, we did not even bother anymore to give names to all the 17 different topological relationships. Figure 2 is a visualization of these relationships.

A similar analysis for the other groups of topological relationships results in a total of 52 real cases (see Table 3).

Table 3. A summary of the analysis for all relationship groups

relationship groups	# possible cases	# real cases
area/area	24	9
line/area	24	17
point/area	4	3
line/line	24	18
point/line	4	3
point/point	2	2
		Grand total 52

4 The Calculus-based Method

The grand total of 52 relationships is far too much for humans to be used in a reasonable manner. It is better to have an overloaded set of just a few basic relationships which the user understands well. The dimension extended method uses various results of feature intersections (empty, 0D, 1D, and 2D) together with the

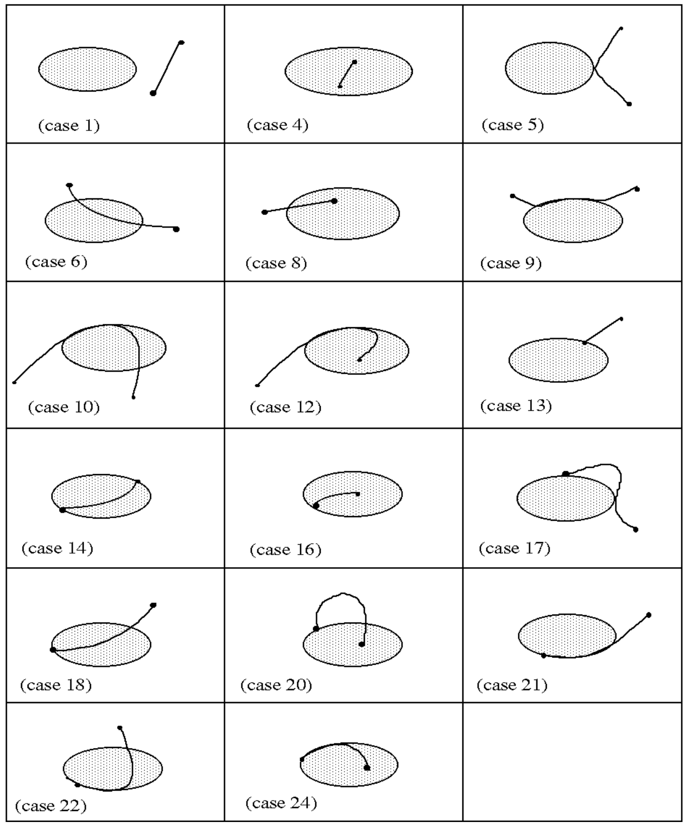

Fig. 2. The 17 different line/area cases in the dimension extended method

boundary and interior operators to describe the required relationships. It may be clear that it is not a very user-friendly method, as the user is not (directly) interested in the intersections of the boundaries and the interiors. Furthermore, though the concept of boundary may be familiar to users, the concept of interior may be less well understood because it is based on the mathematical point-set theory (open/closed sets).

At the query language level, we take into account the considerations above by making available to the users only boundary operators (for area and line features) together with the five topological relationships: *touch*, *in*, *cross*, *overlap*, and *disjoint*. Therefore, in the generic object-calculus fact $\langle \lambda_1, r, \lambda_2 \rangle$, r may be one of the five relationships, while λ_1 and λ_2 may be either features or boundaries of features. We refer to the use of such operators and relationships as the calculus-based method. Formal definitions of these terms will be given in the next subsection. The definitions are general in the sense that they apply to point, line, and area features (unless stated otherwise). It is our conjecture that this is the smallest set of relationships capable of representing all cases of the dimension extended method under the condition that only the additional boundary operators for area and line features are available. The set of topological relationships is close to the normal human use of these concepts and still powerful enough to represent a wide variety of cases.

Based on the formal definitions of the relationships we will prove that they are mutually exclusive and they constitute a full covering of all topological situations. Further, we will give a proof of the fact that all cases of the dimension extended method can be described. Also, a few examples will show that these relationships are capable of distinguishing even more cases (which cannot be described with the dimension extended method).

4.1 Definition of relationships and operators

In the following, an Object-Calculus fact involving a topological relationship is on the left side of the equivalence sign and its definition in the form of a point-set expression is given on the right side.

Definition 1. The *touch* relationship (it applies to area/area, line/line, line/area, point/area, point/line situations, but not to the point/point situation):

$$\langle \lambda_1, touch, \lambda_2 \rangle \Leftrightarrow (\lambda_1^\circ \cap \lambda_2^\circ = \emptyset) \wedge (\lambda_1 \cap \lambda_2 \neq \emptyset) \ .$$

Definition 2. The *in* relationship (it applies to every situation):

$$\langle \lambda_1, in, \lambda_2 \rangle \Leftrightarrow (\lambda_1 \cap \lambda_2 = \lambda_1) \wedge (\lambda_1^\circ \cap \lambda_2^\circ \neq \emptyset) \ .$$

Definition 3. The *cross* relationship (it applies to line/line and line/area situations):

$$\begin{aligned} \langle \lambda_1, cross, \lambda_2 \rangle \Leftrightarrow\ & dim(\lambda_1^\circ \cap \lambda_2^\circ) = (max(dim(\lambda_1^\circ), dim(\lambda_2^\circ)) - 1) \wedge \\ & (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2) \ . \end{aligned}$$

Definition 4. The *overlap* relationship (it applies to area/area and line/line situations):

$$\begin{aligned} \langle \lambda_1, overlap, \lambda_2 \rangle \Leftrightarrow\ & (dim(\lambda_1^\circ) = dim(\lambda_2^\circ) = dim(\lambda_1^\circ \cap \lambda_2^\circ)) \wedge \\ & (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2) \ . \end{aligned}$$

Definition 5. The *disjoint* relationship (it applies to every situation):

$$\langle \lambda_1, disjoint, \lambda_2 \rangle \Leftrightarrow \lambda_1 \cap \lambda_2 = \emptyset \ .$$

A relationship r is symmetric if and only if $\langle \lambda_1, r, \lambda_2 \rangle \Leftrightarrow \langle \lambda_2, r, \lambda_1 \rangle$. A relationship r is transitive if and only if $\langle \lambda_1, r, \lambda_2 \rangle \wedge \langle \lambda_2, r, \lambda_3 \rangle \Rightarrow \langle \lambda_1, r, \lambda_3 \rangle$. It comes from the definitions that all relationships are symmetric with the exception of the *in* relationship. It can be easily proved that only the *in* relationship is transitive.

In order to enhance the use of the above relationships, we define operators able to extract boundaries from areas and lines. With regard to a non-circular line, the boundary ∂L is a set made up of two separate points. Since the 0-dimensional features that we consider are limited to single points, we need to have operators able to access each end-point. We call the end-points f (from) and t (to) respectively, though we do not consider a direction on the line.

Definition 6. The boundary operator b for an area A: The pair (A, b) returns the circular line ∂A.

Definition 7. The boundary operators f, t for a non-circular line L: The pairs (L, f) and (L, t) return the two separate points belonging to the set ∂L.

4.2 Examples

An important advantage of this approach is to provide relationship names that have a reasonably intuitive meaning for users of spatial applications. In the following, we try to substantiate such a claim through several examples.

The *touch* relationship

Intuitively, we say that two geometric elements *touch* each other, if the only thing they have in common is contained in the union of their boundaries. It may be verified easily that all cases in Fig.3 are covered by the formal definition of the *touch* relationship.

The *in* relationship

One feature is *in* another one if the former is completely contained in the latter. The examples of Fig.4 illustrate this relationship.

The *cross* relationship

We say that two lines *cross* each other if they meet on an internal point (note that it could not be a *touch* because in that case the intersection is only on the boundaries). Similarly, a line crosses an area if the line is partly inside the area and partly outside. See Fig.5 for examples of the *cross* relationship.

The *overlap* relationship

Informally, two features *overlap* each other if the result of their intersection is a third feature of the same dimension, but different from both of them. It comes from the definition that this relationship can apply only to homogeneous cases (area/area and line/line, see Fig.6 for a visualization of these cases).

The *disjoint* relationship

Two features are *disjoint* if their intersection is void; this case is quite obvious to understand: see the examples in Fig.7.

4.3 Mutual exclusiveness and full covering of relationships

In this section, we will prove that the five relationships are mutually exclusive, that is, it cannot be the case that two different relationships hold between two features; furthermore, we will prove that they make a full covering of all possible topological situations, that is, given two features, the relationship between them must be one of the five.

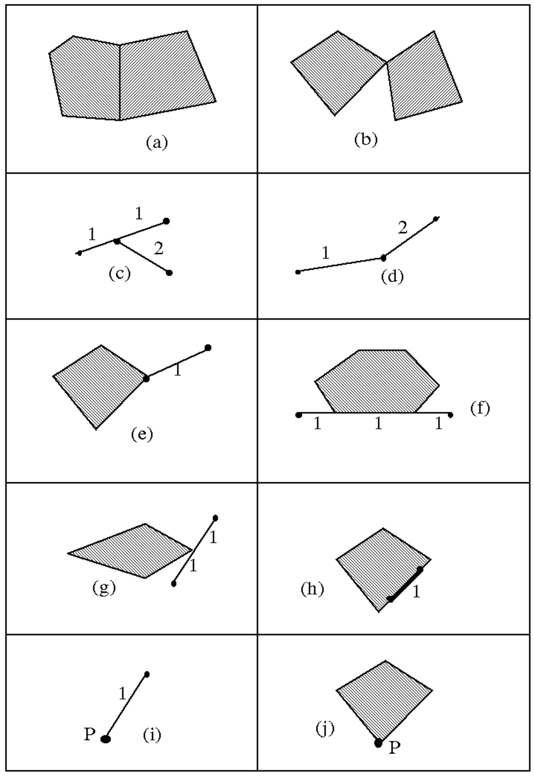

Fig. 3. Topological situations illustrating the *touch* relationship between two areas (a,b), two lines (c,d), a line and an area (e–h), a point and a line (i), a point and an area (j)

Theorem 1. *Given two geometric entities λ_1 , λ_2 and a relationship r between them, if $\langle\lambda_1, r, \lambda_2\rangle$ holds, then $\langle\lambda_1, r_i, \lambda_2\rangle$ does not hold for every $r_i \neq r$, and there does not exist a topological situation that falls outside the five relationships of the calculus-based method.*

Proof. Every internal node (see Fig.8) in the "topological relationship decision" tree represents a boolean predicate; if for a certain topological situation, the predicate evaluates to "true" then the left branch is followed, otherwise the right branch is followed. This process is repeated until a leaf node is reached which will indicate to which of the 5 (or 6 if the asymmetric *in* is counted for two different relationships) basic relationships this situation belongs. Now, two different relationships cannot hold between two given features, because there is only one path to be taken in the topological relationship decision tree. Furthermore, there can be no cases outside the calculus-based method, because (a)

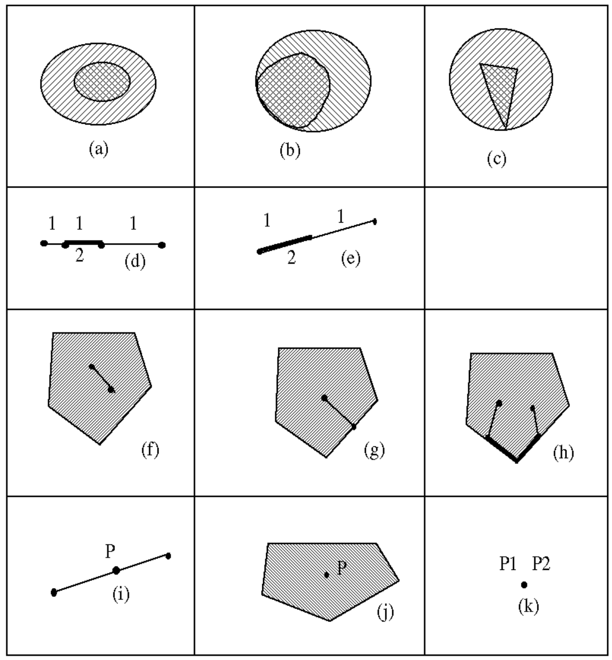

Fig. 4. Topological situations illustrating the *in* relationship between two areas (a–c), two lines (d,e), a line and an area (f–h), a point and a line (i), a point and an area (j), two points (k)

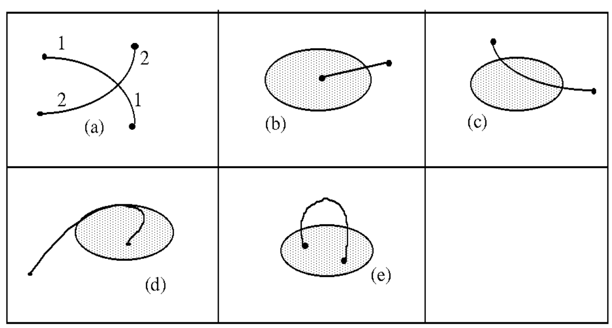

Fig. 5. Topological situations illustrating the *cross* relationship between two lines (a), a line and an area (b–e)

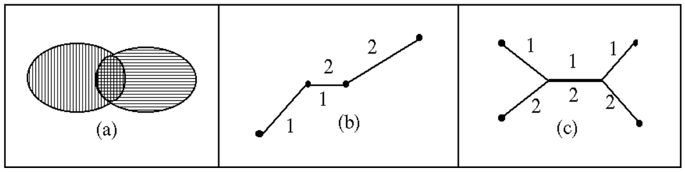

Fig. 6. Topological situations illustrating the *overlap* relationship between two areas (a), two lines (b,c)

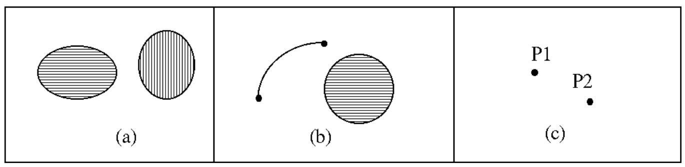

Fig. 7. Topological situations illustrating the *disjoint* relationship between two areas (a), a line and an area (b), two points (c)

every internal node has two branches, so for every value of the predicate there is an appropriate path; and (b) every leaf node has a label that corresponds to one of the five topological relationships. □

Note that the "topological relationship decision" tree is a general tree that can be used for all situations: area/area, line/area, point/area, line/line, point/line, and point/point. From the definition of a point and the predicates it follows that a point can never "travel down" the decision tree below the second level. At this level the relationship (either *touch*, *disjoint*, or *in*) is decided on. In order to evaluate the predicate at the lowest level, one has to take into account the following situations: area/area, line/area, and line/line, because of the use of the dimension function dim in the predicate.

4.4 The calculus-based method versus the dimension extended method

Theorem 2. *The calculus-based method is expressive enough to represent all the topological situations of the dimension extended method.*

Proof. The proof is based on the principle that if we can provide the equivalents of each of the basic terms in the dimension extended method, then we can also specify every case exactly by the logical conjunction ($\wedge$) of these terms. The conjunction of the 4 separate terms will usually result in a quite long expression.

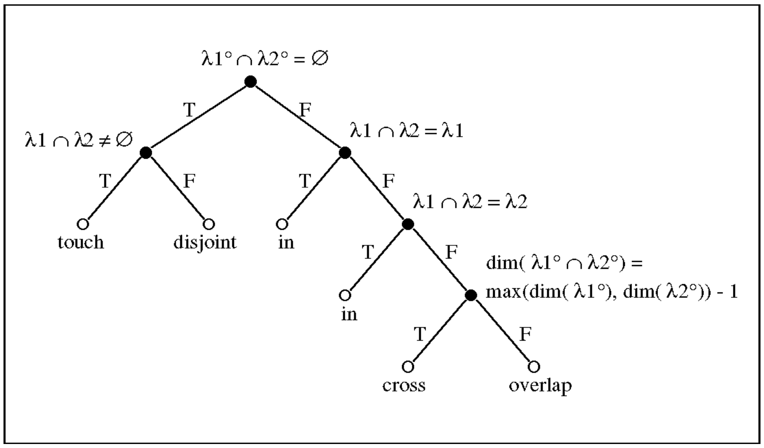

Fig. 8. The topological relationships decision tree

After the proof we will give a few examples showing that the same case can also be specified with a shorter expression.

Each case of the dimension extended method can be specified by the logical conjunction of four terms expressing conditions on the intersection of the boundaries and the interiors of the two features; in general:

$$T_1(\partial\lambda_1 \cap \partial\lambda_2) \wedge T_2(\partial\lambda_1 \cap \lambda_2^\circ) \wedge T_3(\lambda_1^\circ \cap \partial\lambda_2) \wedge T_4(\lambda_1^\circ \cap \lambda_2^\circ) \ . \tag{1}$$

It is possible to give the equivalences for every term T_i admissible in the dimension extended method. On the right of each equivalence we have a logic expression P_i making use of the five relationships between features and between their boundaries. Each equivalence can be easily tested by applying the definitions given for the five relationships. By substituting each T_i with the corresponding P_i, we obtain an expression $P_1 \wedge P_2 \wedge P_3 \wedge P_4$ that is equivalent to (1). Therefore, the calculus-based method is able to express each situation of the dimension extended method. □

In the following, for each term of the dimension extended method, an equivalent term in the calculus-based method is given:

Area/area

$$\begin{aligned}
&\partial A_1 \cap \partial A_2 = \emptyset && \Leftrightarrow \langle (A_1, b), disjoint, (A_2, b)\rangle \\
&dim(\partial A_1 \cap \partial A_2) = 0 && \Leftrightarrow \langle (A_1, b), cross, (A_2, b)\rangle \\
&dim(\partial A_1 \cap \partial A_2) = 1 && \Leftrightarrow \langle (A_1, b), overlap, (A_2, b)\rangle \vee \langle (A_1, b), in, (A_2, b)\rangle
\end{aligned}$$

$$
\begin{array}{ll}
\partial A_1 \cap A_2^\circ = \emptyset & \Leftrightarrow \langle A_2, in, A_1\rangle \vee \langle A_2, touch, A_1\rangle \vee \langle A_2, disjoint, A_1\rangle \\
dim(\partial A_1 \cap A_2^\circ) = 1 & \Leftrightarrow (\langle A_1, in, A_2\rangle \wedge (\langle (A_1, b), disjoint, (A_2, b)\rangle \vee \\
 & \quad \langle (A_1, b), cross, (A_2, b)\rangle \vee \\
 & \quad \langle (A_1, b), overlap, (A_2, b)\rangle)) \vee \langle A_1, overlap, A_2\rangle \\
A_1^\circ \cap \partial A_2 = \emptyset & \Leftrightarrow \langle A_1, in, A_2\rangle \vee \langle A_1, touch, A_2\rangle \vee \langle A_1, disjoint, A_2\rangle \\
dim(A_1^\circ \cap \partial A_2) = 1 & \Leftrightarrow (\langle A_2, in, A_1\rangle \wedge (\langle (A_2, b), disjoint, (A_1, b)\rangle \vee \\
 & \quad \langle (A_2, b), cross, (A_1, b)\rangle \vee \\
 & \quad \langle (A_2, b), overlap, (A_1, b)\rangle)) \vee \langle A_2, overlap, A_1\rangle \\
A_1^\circ \cap A_2^\circ = \emptyset & \Leftrightarrow \langle A_1, touch, A_2\rangle \vee \langle A_1, disjoint, A_2\rangle \\
dim(A_1^\circ \cap A_2^\circ) = 2 & \Leftrightarrow \langle A_1, in, A_2\rangle \vee \langle A_2, in, A_1\rangle \vee \langle A_1, overlap, A_2\rangle
\end{array}
$$

Line/line

$$
\begin{array}{ll}
\partial L_1 \cap \partial L_2 = \emptyset & \Leftrightarrow \langle (L_1, f), disjoint, (L_2, f)\rangle \wedge \\
 & \quad (\langle (L_1, t), disjoint, (L_2, f)\rangle \wedge \\
 & \quad \langle (L_1, f), disjoint, (L_2, t)\rangle \wedge \\
 & \quad (\langle (L_1, t), disjoint, (L_2, t)\rangle \\
dim(\partial L_1 \cap \partial L_2) = 0 & \Leftrightarrow \langle (L_1, f), touch, L_2\rangle \vee \langle (L_1, t), touch, L_2\rangle \\
\partial L_1 \cap L_2^\circ = \emptyset & \Leftrightarrow (\langle (L_1, f), disjoint, L_2\rangle \vee \langle (L_1, f), touch, L_2\rangle) \wedge \\
 & \quad (\langle (L_1, t), disjoint, L_2\rangle \vee \langle (L_1, t), touch, L_2\rangle) \\
dim(\partial L_1 \cap L_2^\circ) = 0 & \Leftrightarrow \langle (L_1, f), in, L_2\rangle \vee \langle (L_1, t), in, L_2\rangle \\
L_1^\circ \cap \partial L_2 = \emptyset & \Leftrightarrow (\langle (L_2, f), disjoint, L_1\rangle \vee \langle (L_2, f), touch, L_1\rangle) \wedge \\
 & \quad (\langle (L_2, t), disjoint, L_1\rangle \vee \langle (L_2, t), touch, L_1\rangle) \\
dim(L_1^\circ \cap \partial L_2) = 0 & \Leftrightarrow \langle (L_2, f), in, L_1\rangle \vee \langle (L_2, t), in, L_1\rangle \\
L_1^\circ \cap L_2^\circ = \emptyset & \Leftrightarrow \langle L_1, disjoint, L_2\rangle \vee \langle L_1, touch, L_2\rangle \\
dim(L_1^\circ \cap L_2^\circ) = 0 & \Leftrightarrow \langle L_1, cross, L_2\rangle \\
dim(L_1^\circ \cap L_2^\circ) = 1 & \Leftrightarrow \langle L_1, overlap, L_2\rangle \vee \langle L_1, in, L_2\rangle \vee \langle L_2, in, L_1\rangle
\end{array}
$$

Line/area

$$
\begin{array}{ll}
\partial A \cap \partial L = \emptyset & \Leftrightarrow \langle (L, f), disjoint, (A, b)\rangle \wedge \langle (L, t), disjoint, (A, b)\rangle \\
dim(\partial A \cap \partial L) = 0 & \Leftrightarrow \langle (L, f), in, (A, b)\rangle \vee (\langle (L, t), in, (A, b)\rangle \\
\partial A \cap L^\circ = \emptyset & \Leftrightarrow \langle L, disjoint, (A, b)\rangle \vee \langle L, touch, (A, b)\rangle \\
dim(\partial A \cap L^\circ) = 0 & \Leftrightarrow \langle L, cross, (A, b)\rangle \\
dim(\partial A \cap L^\circ) = 1 & \Leftrightarrow \langle L, overlap, (A, b)\rangle \vee \langle L, in, (A, b)\rangle \\
A^\circ \cap \partial L = \emptyset & \Leftrightarrow (\langle (L, f), disjoint, A\rangle \vee \langle (L, f), touch, A\rangle) \wedge \\
 & \quad (\langle (L, t), disjoint, A\rangle \vee \langle (L, t), touch, A\rangle) \\
dim(A^\circ \cap \partial L) = 0 & \Leftrightarrow \langle (L, f), in, A\rangle \vee \langle (L, t), in, A\rangle \\
A^\circ \cap L^\circ = \emptyset & \Leftrightarrow \langle L, touch, A\rangle \vee \langle L, disjoint, A\rangle \\
dim(A^\circ \cap L^\circ) = 1 & \Leftrightarrow \langle L, cross, A\rangle \vee \langle L, in, A\rangle
\end{array}
$$

Point/line

$$
\begin{array}{ll}
\partial L \cap P = \emptyset & \Leftrightarrow \langle P, disjoint, L\rangle \vee \langle P, in, L\rangle \\
dim(\partial L \cap P) = 0 & \Leftrightarrow \langle P, touch, L\rangle \\
L^\circ \cap P = \emptyset & \Leftrightarrow \langle P, disjoint, L\rangle \vee \langle P, touch, L\rangle \\
dim(L^\circ \cap P) = 0 & \Leftrightarrow \langle P, in, L\rangle
\end{array}
$$

Point/area

$$\begin{array}{ll} \partial A \cap P = \emptyset & \Leftrightarrow \langle P, disjoint, A\rangle \vee \langle P, in, A\rangle \\ dim(\partial A \cap P) = 0 & \Leftrightarrow \langle P, touch, A\rangle \\ A^\circ \cap P = \emptyset & \Leftrightarrow \langle P, disjoint, A\rangle \vee \langle P, touch, A\rangle \\ dim(A^\circ \cap P) = 0 & \Leftrightarrow \langle P, in, A\rangle \end{array}$$

Point/point

$$\begin{array}{ll} P_1 \cap P_2 = \emptyset & \Leftrightarrow \langle P_1, disjoint, P_2\rangle \\ dim(P_1 \cap P_2) = 0 & \Leftrightarrow \langle P_1, in, P_2\rangle \end{array}$$

An example may help to understand the proof: let us consider case 5 of Table 2, which is expressed by:

$$(\partial A \cap \partial L = \emptyset) \wedge (dim(\partial A \cap L^\circ) = 0) \wedge (A^\circ \cap \partial L = \emptyset) \wedge (A^\circ \cap L^\circ = \emptyset);$$

by making all the substitutions with the equivalences given above, it can be expressed by:

$$\begin{array}{l} \langle (L,f), disjoint, (A,b)\rangle \wedge \langle (L,t), disjoint, (A,b)\rangle \wedge \\ \langle L, cross, (A,b)\rangle \wedge \\ (\langle (L,f), disjoint, A\rangle \vee \langle (L,f), touch, A\rangle) \wedge \\ (\langle (L,t), disjoint, A\rangle \vee \langle (L,t), touch, A\rangle) \wedge \\ \langle L, touch, A\rangle \vee \langle L, disjoint, A\rangle \ . \end{array}$$

Of course, this is a long expression, valid in general, but not so practical. An TTad hoc" expression much more effective for the same case is the following:

$$\langle L, touch, A\rangle \wedge \langle L, cross, (A,b)\rangle \wedge \langle (L,f), disjoint, A\rangle \wedge \langle (L,t), disjoint, A\rangle.$$

Other examples of some situations in Fig.3 are simply expressed by:

$$\begin{array}{ll} \langle A_1, touch, A_2\rangle \wedge \langle (A_1,b), overlap, (A_2,b)\rangle & \text{(Fig.3.a)} \\ \langle A_1, touch, A_2\rangle \wedge \langle (A_1,b), cross, (A_2,b)\rangle & \text{(Fig.3.b);} \end{array}$$

and some situations in Fig.4 by:

$$\begin{array}{ll} \langle (L_2, in, L_1)\rangle \wedge \langle (L_2,f), in, L_1)\rangle \wedge \langle (L_2,t), in, L_1)\rangle & \text{(Fig.4.d)} \\ \langle L, in, A\rangle \wedge \langle L, overlap, (A,b)\rangle \wedge \langle (L,f), in, A\rangle \wedge \langle (L,t), in, A\rangle & \text{(Fig.4.h).} \end{array}$$

Theorem 2 states that all the cases in the dimension extended method can be expressed with the calculus-based method. But is the converse true? It is easy to see that there are some topological situations that are undistinguishable in the dimension extended method, but that can be represented with the calculus-based method.

For example, the two situations between the lines L_1 and L_2 in Fig.9.a both fall in the following case in the dimension extended method:

$$(\partial L_1 \cap \partial L_2 = \emptyset) \wedge (\partial L_1 \cap L_2^\circ = \emptyset) \wedge (dim(L_1^\circ \cap \partial L_2) = 0) \wedge (L_1^\circ \cap L_2^\circ = \emptyset),$$

while we can make a distinction with the primitives of the calculus-based method:

$I.\ \langle L_1, touch, L_2\rangle \wedge ((\langle (L_2, f), in, L_1\rangle \wedge \langle (L_2, t), disjoint, L_1\rangle)\vee$
$(\langle (L_2, t), in, L_1\rangle \wedge \langle (L_2, f), disjoint, L_1\rangle));$
$II.\ \langle L_1, touch, L_2\rangle \wedge \langle (L_2, f), in, L_1\rangle \wedge \langle (L_2, t), in, L_1\rangle.$

Another example is depicted in Fig.9.b, where both situations correspond to the line/area case no. 20 in Table 2. The first situation is a *cross*, while the second one is a in; in detail:

$I.\ \langle L, cross, A\rangle \wedge \langle L, cross, (A, b)\rangle \wedge ((\langle\langle (L, f), in, (A, b)\rangle \wedge (\langle (L, t), in, A\rangle)\rangle\vee$
$(\langle (L, t), in, (A, b)\rangle \wedge (\langle (L, f), in, A\rangle)\rangle);$
$II.\ \langle L, in, A\rangle \wedge \langle L, cross, (A, b)\rangle \wedge ((\langle\langle (L, f), in, (A, b)\rangle \wedge (\langle (L, t), in, A\rangle)\rangle\vee$
$(\langle (L, t), in, (A, b)\rangle \wedge (\langle (L, f), in, A\rangle)\rangle).$

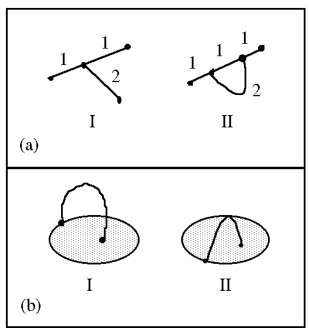

Fig. 9. Comparison between the calculus-based method and the dimension extended method

This additional expressive power comes with the *in* relationship and the f and t operators. In fact, the *in* relationship allows to say (see Def.2) that the result of the intersection of the two entities is equal to one of them (not only the dimension of the result like in the dimension extended method); furthermore, the f and t operators allow to specify conditions on the single end-point of a line (in the dimension extended method, the boundary of a line is a unitary concept).

5 Discussion

In conclusion, we proposed a formal way of modeling topological relationships adopting a calculus-based method, suitable for the definition of an actual query language towards GISs, and close to the way users think about topological relationships. We defined the calculus-based method starting from a point-set theory approach, which is the recent one adopted in the literature (e.g. [EF91]) to model topological situations.

The cases that are left out during this first presentation of the calculus-based method are:

1. complex area or line features (that is, in case of an area: a non-connected boundary, and in case of a line: more than two end-points and self-intersections) (Fig.10.a-b);
2. distinguishing the number of simple features that can characterize the intersection of two features (Fig.10.c-d);
3. ordering of the different parts of crossing lines (Fig.10.e-f);
4. line features crossing above/below each-other; this is left out because it is a 3D problem (we will need a 3D variant of the method) (Fig.10.g).

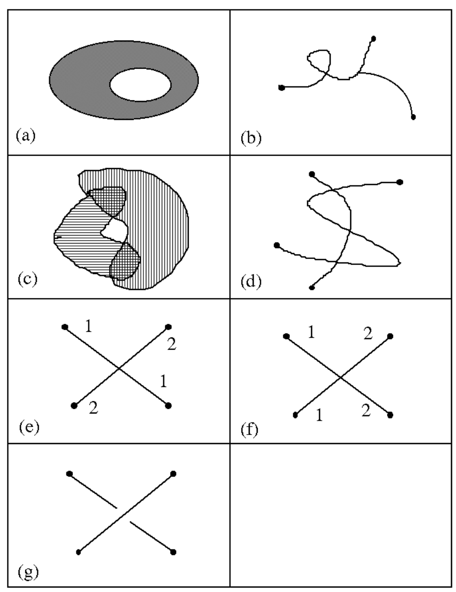

Fig. 10. Extensions left out in our modeling

We plan to extend the calculus-based method to encompass also the cases above. Another point in our wish list is related to test if the calculus-based method is really suitable for end-users. This will lead to some experiments on human subjects to check if the way we grouped topological situations is close to the way people do the same and, therefore, check the usefulness of our query language.

The proposed operators b, f, and t (boundary, from, and to respectively) and the topological relationships have been all implemented as functions (and operators) in the Postgres [SR86, SRH90] extendible DBMS environment on a Sun Sparc. It is implemented in a manner similar to (and compatible with) the standard geometric extension used in GEO++ [vOV91, VvO92]. When using the topological relationships in the Postgres/GEO++ environment, one should be aware that due to the Postquel query language, the syntax is a little different from the Object-Calculus. However, the semantics of the implemented relationships and methods are exactly the same.

Acknowledgments

This work has been supported by "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" of the Italian National Council of Research (C.N.R.) under grant no. 92.01574.PF69 and by the division "Software Development and Information Technology" of the TNO Physics and Electronics Laboratory.

Special thanks go to Max Egenhofer for his help and suggestions.

References

[Abl87] Ronald F. Abler. The national science foundation national center for geographic information and analysis. *International Journal of Geographical Information Systems*, 1(4):303–326, 1987.

[B+91] K. Bennis et al. GéoGraph: A topological storage model for extensible GIS. In *Auto-Carto 10*, pages 349–367, March 1991.

[CD92] Eliseo Clementini and Paolino Di Felice. An object calculus for geographic databases. Technical report, Dipartimento di Ingegneria Elettrica, Universitá dell'Aquila, Italy, 1992. submitted for publication.

[CDD91] Eliseo Clementini, Paolino Di Felice, and Alessandro D'Atri. A spatial data model underlying human interaction with object-oriented spatial databases. In *Fifteenth Annual International Computer Software & Applications Conference*, pages 110–117, Tokyo, September 1991. IEEE Computer Society Press.

[D+90] O. Deux et al. The story of O_2. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):91–108, March 1990.

[dHvO92] Sylvia de Hoop and Peter van Oosterom. Storage and manipulation of topology in Postgres. In *Third European Conference on Geographical Information Systems*, pages 1324–1336, Munich, March 1992.

[EF91] Max J. Egenhofer and Robert D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

[EH92] Max J. Egenhofer and John R. Herring. Categorizing binary topological relationships between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono, ME, 1992. submitted for publication.

[Kel55] John L. Kelley, editor. *General Topology*. Springer-Verlag, New York, 1955.

[ME92] David M. Mark and Max J. Egenhofer. An evaluation of the 9-intersection for region-line relations. In *GIS/LIS Conference*, San Jose, CA, November 1992.

[MS89] Sudhakar Menon and Terence R. Smith. A declarative spatial query processor for Geographic Information Systems. *Photogrammetric Engineering and Remote Sensing*, 55(11):1593–1600, November 1989.

[RFS88] Nick Roussopoulos, Christos Faloutsos, and Timos Sellis. An efficient pictorial database system for PSQL. *IEEE Transactions on Software Engineering*, 14(5):639–650, May 1988.

[SR86] Michael Stonebraker and Lawrence A. Rowe. The design of Postgres. *ACM SIGMOD*, 15(2):340–355, 1986.

[SRH90] Michael Stonebraker, Lawrence A. Rowe, and Michael Hirohama. The implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142, March 1990.

[vOV91] Peter van Oosterom and Tom Vijlbrief. Building a GIS on top of the open DBMS "Postgres". In *Proceedings EGIS'91: Second European Conference on Geographical Information Systems*, pages 775–787, Utrecht, April 1991. EGIS Foundation.

[VvO92] Tom Vijlbrief and Peter van Oosterom. The GEO system: An extensible GIS. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, pages 40–50, Charleston, South Carolina, August 1992. International Geographical Union IGU.

This article was processed using the LaTeX macro package with LLNCS style