$\mathcal{I}$-SQE: A Query Engine for Answering Range Queries over Incomplete Spatial Databases

Alfredo Cuzzocrea[1] and Andrea Nucita[2]

[1] ICAR-CNR and University of Calabria, Italy
cuzzocrea@si.deis.unical.it
[2] University of Messina, Italy
andrea@informatica.unime.it

Abstract. Spatial database systems built on top of distributed and heterogeneous spatial information sources such as conventional spatial databases underlying *Geographical Information Systems* (GIS), spatial data files and spatial information acquired or inferred from the Web, suffer from *data integration* and *topological consistency* problems. These issues make the globally-integrated spatial database *incomplete*, so that effectively and efficiently answering range queries over such databases represents a leading challenge for spatial database systems research. Inspired by these motivations, in this paper we propose $\mathcal{I}$-SQE (*Spatial Query Engine for $\mathcal{I}$ncomplete information*), an innovative query engine for answering range queries over incomplete spatial databases via meaningfully integrating *geometrical information* and *topological reasoning*. $\mathcal{I}$-SQE finally allows us to enhance the quality and the expressive power of retrieved answers by meaningfully taking advantages from the amenity of representing spatial database objects via both the geometrical and the topological level.

1 Introduction

In modern spatial database environments, data repositories collected and integrated from different spatial information sources very often coexist. Conventional spatial databases such as those underlying autonomous *Geographical Information Systems* (GIS), raw data files storing geographical information, and GIS-related Web pages are popular instances of these sources. Furthermore, the proliferation of Web- and Grid-service-based applications and systems built on top of spatial data repositories leads to an Internet-wide dissemination of spatial information sources. This phenomenon makes dealing with integration issues of spatial database systems more difficult. While the popularity of spatial data repositories within modern complex information systems, such as Web, Grid and P2P systems, is clearly an opportunity that puts the basis for further studies in the field and, symmetrically, for the industrial proliferation of spatial database systems, spatial data repositories collected and integrated from different spatial information sources also pose several research challenges. These challenges mainly concern with the presence of *incomplete information* [14,3,7] due to the

J.D. Velásquez et al. (Eds.): KES 2009, Part II, LNAI 5712, pp. 91–101, 2009.

heterogeneity of spatial data repositories according to several aspects, such as data models, data formats, ranges of data domains, null values handling policies, and so forth.

In heterogeneous spatial database environments like those illustrated above, *data integration* is the first issue to be faced-off [5]. Data integration has been extensively studied in the context of spatial databases (e.g., [6,10]). On the other hand, another leading challenge in spatial database systems research is represented by the issue of extending the capabilities of conventional query engines in order to make them able of dealing with the presence of several *heterogenous representations* of spatial information (e.g., [18]), which very often arise in actual GIS. This paradigm pursues the idea of representing the *same* spatial information kept in *spatial database objects* according to different levels, or *layers*, in order to enhance the expressive power of both abstraction and reasoning capabilities over spatial data. It should be noted that the latter one is a critical aspect in spatial database systems research, as modern complex information systems are more and more heterogenous in nature and kind of underlying data repositories, so that heterogeneous representations of spatial information arise accordingly. As a consequence, spatial query engines interfacing these systems have to cope with the deriving data integration issues.

Inspired by these considerations, in this paper we present $\mathcal{I}$-SQE (*Spatial Query Engine for $\mathcal{I}$ncomplete information*), a query engine for answering range queries over incomplete spatial databases, like those that derive from integrating distributed and heterogeneous spatial information sources. In particular, in the context of $\mathcal{I}$-SQE we investigate the problem of answering range queries over spatial databases where spatial information is modeled and represented according to two different levels, i.e. the *geometrical level* and the *topological level*, respectively. Also, for a sub-set of spatial database objects stored in the target spatial database interfaced by $\mathcal{I}$-SQE, one of these two levels can be missing, so that, as a consequence, incomplete spatial information occurs, and the spatial database is incomplete. The main goal of $\mathcal{I}$-SQE consists in devising intelligent techniques for answering range queries over this kind of spatial databases while overcoming incompleteness limitations.

In a conventional spatial database, spatial information is usually represented by means of detailed *geometrical properties* of spatial database objects. This because geometrical one is the most complete representation one can provide about spatial database objects. For instance, given a collection of spatial database objects, *topological relations* among these objects (e.g., containment relations) can be derived from their geometrical properties. In $\mathcal{I}$-SQE, we consider an application scenario where spatial information can be incomplete, i.e. a sub-set of spatial database objects is described by their topological relations with other spatial database objects stored in the target spatial database, whereas the geometrical information about these objects is missing. As a consequence, conventional spatial query engines, which are based on the complete availability of geometrical information about spatial database objects, are not able of effectively and efficiently answering spatial queries involving such objects.

To give an example, consider the simple case of a spatial database representing streets of a given urban area, along their geometry (i.e., geometrical information is available). Furthermore, assume that the spatial database also stores topological relations about regional areas and streets, while the geometry of regional areas is not known (i.e., topological information is available while geometrical information is not available). The simplest case of topological relation is represented by the containment one, which models the fact that a regional area A contains a set of streets $\{S_0, S_1, \ldots, S_{N-1}\}$, such that $N > 0$. If only the geometrical layer is exploited to answer range queries over the spatial database, then users only retrieve geometrical information about streets, whereas topological information on regional areas is not exploited at all. It should be noted that, in a scenario like the one described above, knowledge extracted from topological relations represents a critical "add-in" value for modern GIS applications and systems, as this information can be further exploited to enhance the knowledge discovery phase from spatial databases.

Contrary to the example above, in $\mathcal{I}$-SQE users are allowed to integrate knowledge kept in both levels, i.e. the geometrical and the topological level, respectively, thus taking advantages from both the different data representation models. Moreover, it should be noted that this paradigm is also "self-alimenting", meaning that new topological relations among queried spatial database objects can be derived by means of simple yet effective *composition rules* over already-extracted topological relations made available in the spatial database system via the query task.

The remaining part of this paper is organized as follows. In Sect. 2, we briefly review research efforts related to our research. Sect. 3 describes our technique for answering range queries over incomplete spatial databases via integrating geometrical information and topological reasoning. In Sect. 4, we present in detail $\mathcal{I}$-SQE, along its main principles, components and reference architecture. Finally, Sect. 5 discusses conclusions and future work of our research.

2 Related Work

In recent years, the proliferation of spatial data repositories has posed several challenges related to data integration issues from distributed and heterogeneous spatial information sources. For instance, the huge quantity of spatial data available on the Web leads to the possibility of their acquisition and integration within GIS, also in a semi-automatic manner [17]. Nevertheless, methods for spatial data acquisition are manyfold, and each GIS software makes use of different and heterogenous formats for representing spatial data. As a consequence, inconsistency and incompleteness arise in merged spatial data repositories, and a reasonable solution to these issues is represented by data integration techniques over such repositories.

In [13], authors propose a system for spatial data integration and sharing throughout Web services technology, via using standard Web languages and protocols such as *Geography Markup Language* (GML) [1] and *Simple Object*

Access Protocol (SOAP) [2]. In [9], a method able of evaluating queries over integrated spatial database systems is presented. Given an input spatial query Q, this method finds an optimal query execution plan for Q from the different plans computed for each feature of the integrated spatial data repository.

Models assuming the presence of *different representation layers* for spatial information have been introduced in past research efforts. In [3], a model that integrates multiple representations of *geographical maps* is presented. This model is called *Layered Spatial Data Model* (LSDM). The peculiarity of LSDM relies in the ability of representing *incomplete maps*, i.e. maps for which the geometry of the contained objects is not completely known. In addition to this, in LSDM it is also possible to represent combinatorial and topological relations among spatial database objects that qualitatively represent maps regardless of geometrical properties needed to compute them.

On the other hand, the wide availability of multi-level representation models, multi-resolution maps and spatial data mined from the Web [17] imposes us new challenges with respect to check and validation of consistency of topological relations in a spatial database system. Following this fundamental issue, [4] introduces a model for evaluating the consistency of topological relations when multi-resolution maps built on top of spatial databases are considered. As studied in [4], the main problem to be faced-off in this case relies on the fact that, in collections of multi-resolution maps that one can find in a GIS, the same spatial database object could be represented at various resolutions in different maps. This poses data as well as knowledge integration aspects to be considered.

Without doubts, topological information plays a crucial role in spatial query processing, as its semantics can be further exploited in order to improve the query capabilities of GIS integrating topology-based query engines. Nevertheless, the management of topological relations is space- and time-consuming [14]. As a consequence, devising efficient methods for representing, managing and querying topological relations plays a leading role in spatial query processing of modern GIS architectures.

In line with the considerations above, [11] proposes reducing the number of *false positives* that can be retrieved during the filtering phase of spatial selection queries via equipping each spatial database object stored in nodes of the *R*-tree indexing the spatial database with the so-called *Internal Rectangle* (IR). IRs are used to meaningfully infer topological relations among spatial database objects. For instance, if two IRs overlap, the actual spatial database objects overlap too. Being based on IRs, this method significantly reduces computational overheads due to computing topological relations among objects, and, as a nice consequence, the time needed to answer spatial queries involving these objects.

In [15], authors try to answer the following question: *"Which topological information on actual spatial database objects is possible to infer from topological relations among their respective MBRs?"*. They state that topological information on spatial database objects can be inferred from the *relative positions* of their respective MBRs. This fundamental insight puts the basis towards defining novel optimization strategies for efficient spatial query processing.

Finally, in [16] authors introduce a method for reducing the number of *spatial constraints* of queries via discarding those constraints that can be inferred from a sub-set of the whole (spatial) constraint set. Apart from improving the performance of spatial query evaluation, reducing the number of spatial constraints can be also useful to achieve a more compact storage representation of topological information. In a similar research initiative ([12]), *Multi-Scale Heuler Histograms* (MSHH) are proposed as a new technique for obtaining high-performance compressed representations of topological information.

In all the research initiatives reviewed above, it is always assumed that geometrical information is available for all the spatial database objects stored in the target spatial database. This allows topological relations to be computed from geometrical information in an easy manner. Contrary to this, in our research we address the relevant challenge of answering range queries over spatial databases in the presence of incomplete information, i.e. the case in which a sub-set of spatial database objects are described in the target spatial database by means of topological information only, while geometrical information associated to these objects is missing. This connotes the whole spatial information associated to these objects as incomplete.

3 Integrating Geometrical Information and Topological Reasoning for Answering Range Queries over Incomplete Spatial Databases

In this Section, we present our technique for integrating geometrical information and topological reasoning in order to answer range queries over incomplete spatial databases. This technique is implemented by algorithm `evaluateRangeQuery`, which represents the core layer of $\mathcal{I}$-SQE, our proposed query engine for incomplete spatial databases.

As highlighted in Sect. 1, in $\mathcal{I}$-SQE we focus the attention on the challenging case of dealing with incomplete spatial databases where geometrical information associated to a sub-set of spatial database objects stored in the target spatial database is missing, whereas a topological layer describing topological relations among these objects is available. Let us denote as $\mathcal{D}$ the spatial database. We denote as $G_{\mathcal{D}}$ the set of spatial database objects for which geometrical information is available, and as $T_{\mathcal{D}}$ the set of spatial database objects for which only topological information is available. For the sake of simplicity, we name as *geometrical objects* spatial database objects belonging to $G_{\mathcal{D}}$, whereas as *topological objects* spatial database objects belonging to $T_{\mathcal{D}}$, respectively. Also, we assume that spatial database objects are indexed by means of classical MBRs embedded in a high-performance R-tree indexing data structure, and that input queries are modeled in terms of two-dimensional range (spatial) queries. For instance, a typical spatial query Q belonging to this class of queries could ask if a certain spatial object O is contained by or intersects the range R of Q.

In our reference spatial database scenario, topological information is stored in the target spatial database by means of a simple yet effective two-dimensional

Table 1. A 3×3-array storing topological information on the spatial database objects O_i, O_j, and O_k

	O_i	O_j	O_k
O_i	E	I	O
O_j	Ct	E	I
O_k	Ct	Ct	E

array such that each entry $\langle O_i, O_j \rangle$ contains the topological relation $T_{i,j}$ among the spatial database objects O_i and O_j, i.e. $O_i \cdot T_{i,j} \cdot O_j$. Table 1 shows an example of a 3×3-array storing topological information on the spatial database objects O_i, O_j, and O_k. Here, E denotes the topological relation *Equal*, I *Inside*, O *Overlap*, and Ct *Contain*. For instance, $O_i \cdot I \cdot O_j$ models the topological relation stating that the spatial database object O_i is inside the spatial database object O_j (i.e., O_j contains O_i); $O_i \cdot O \cdot O_k$ means that O_i overlaps O_k, and so forth.

In a conventional spatial database system, topological information can become very large, due to the fact that a huge number of topological relations among spatial database objects can exist, as the number of topological relations is quadratic in the number of spatial database objects. As a consequence, similarly to proper spatial database objects that are indexed via high-performance R-trees, topological relations are indexed via conventional B-trees that are suitable to categorical data, and also embed efficient search algorithms for retrieving the desired information. Therefore, in our reference spatial database scenario we assume that a B-tree indexing topological relations is available.

Given a range query Q over $\mathcal{D}$ involving a set of spatial database objects belonging to $G_{\mathcal{D}} \bigcup T_{\mathcal{D}}$, our goal is to integrate geometrical information and topological information in order to provide an answer to Q, denoted by $A(Q)$. $A(Q)$ is composed by two kinds of objects: (i) geometrical objects in $G_{\mathcal{D}}$ involved by Q, for which topological relations among the geometry of these objects and the range R of Q can be easily computed; (ii) topological objects modeling topological relations between topological objects in $T_{\mathcal{D}}$ involved by Q and the range R of Q, which, contrary to the previous case, must be inferred via the method we propose in this research (recall that, for spatial database objects referred by $T_{\mathcal{D}}$, geometrical information is not available). In more detail, answering Q over $\mathcal{D}$ is performed according to a double-step approach. First, geometrical objects involved by Q are retrieved via the R-tree indexing data structure. Then, topological objects involved by Q are retrieved by means of *compositions* of topological relations between topological objects in $T_{\mathcal{D}}$ and the range R of Q. During this step, the B-tree is exploited to efficiency purposes.

Handling topological information represents a non-trivial engagement. In fact, it should be noted that topological relations retrieved during the evaluation of an input range query Q could be modeled in terms of a *disjunction of* (*basic*) *topological relations*. For instance, given a topological object O and the range R of Q, a possible disjointed expression could be: $O \cdot (Overlap \vee Inside) \cdot R$, which models the fact that O can alternatively overlap R or being contained by R. Hence, we classify the topological objects retrieved by evaluating Q into two

Input: The incomplete spatial database $\mathcal{D}$; the range query Q.
Output: The answer to Q, $A(Q)$.
Method: Perform the following steps:

$A(Q) \leftarrow \langle \emptyset, \emptyset \rangle$;
$\mathcal{A} \leftarrow$ *initializeArray*();
$\mathcal{G} \leftarrow$ *retrieveGeometricalObjects*($\mathcal{D}$,Q);
$\mathcal{A}.add(\mathcal{G})$;
for each g **in** $\mathcal{G}${
  $\mathcal{R} \leftarrow$ *getTopologicalRelations*($\mathcal{D}$,$\mathcal{G}$,g);
  $\mathcal{T} \leftarrow$ *retrieveTopologicalObjects*($\mathcal{D}$,$\mathcal{R}$);
  $\mathcal{A}.add(\mathcal{T})$;
}
$\mathcal{A}.add(Q)$;
$\mathcal{R} \leftarrow$ *computeTopologicalRelations*($\mathcal{A}$);
$\mathcal{T} \leftarrow$ *retrieveTopologicalObjects*(Q,$\mathcal{R}$);
$A(Q) \leftarrow \langle \mathcal{G}, \mathcal{T} \rangle$;
return $A(Q)$;

Fig. 1. Algorithm `evaluateRangeQuery`

possible classes, namely *certain topological objects*, for which topological relations with the range R of Q can be determined exactly, and *uncertain topological objects*, for which topological relations with the range R of Q are described by a disjunction of basic topological relations, i.e. an exact representation cannot be retrieved.

In light of this, building compositions of topological relations in order to retrieve topological objects in $A(Q)$ can be questioning, due to the presence of incompleteness and uncertainty in spatial data. However, some intuitive optimizations can be devised, in order to tame computational overheads introduced by this task during the evaluation of Q. In fact, among all topological objects in $T_{\mathcal{D}}$, those that can be exploited to model compositions to be retrieved with $A(Q)$ are those for which a topological relation different from $Disjoint$ and $Universal$ with *at least* one geometrical object in $G_{\mathcal{D}}$ involved by Q exists. Recall that, given two spatial database objects O_i and O_j, $O_i \cdot Disjoint \cdot O_j$ models the fact that O_i and O_j do not have any spatial point in common (i.e., $O_i \bigcap O_j = \emptyset$), whereas $O_i \cdot Universal \cdot O_j$ models the fact that every topological relation between O_i and O_j can exist, i.e. information about the topological relation between O_i and O_j is null.

Algorithm `evaluateRangeQuery` (see Fig. 1) implements our proposed technique for answering range queries over incomplete spatial databases via integrating geometrical information and topological reasoning. Recall that, in our reference spatial database scenario, we assume that topological information about spatial database objects stored in the target spatial database is already computed and made available. `evaluateRangeQuery` takes as input an incomplete spatial database $\mathcal{D}$ and a range query Q over $\mathcal{D}$, and returns as output the answer to Q, $A(Q)$. In more detail, `evaluateRangeQuery` makes use of the following procedures: (i) `initializeArray`, which initializes the two-dimensional array $\mathcal{A}$ used as a temporary data structure to store topological information in the vest of intermediate results for the answer to Q, $A(Q)$; (ii) `retrieveGeometricalObjects`, which takes as input a spatial database $\mathcal{D}$ and a range query Q over $\mathcal{D}$, and returns as output the set of geometrical objects in $\mathcal{D}$ having a non-null intersection with Q; (iii) `add`, which takes as input a set of geometrical objects $\mathcal{G}$ and, applied to a two-dimensional array $\mathcal{A}$, adds to $\mathcal{A}$ appropriate identifiers of objects in $\mathcal{G}$; (iv) `getTopologicalRelations`, which takes as input a spatial database $\mathcal{D}$, a set of geometrical objects $\mathcal{G}$ and a geometrical object g, and returns as output the set of topological relations between g and geometrical objects in $\mathcal{G}$; (v) `retrieveTopologicalObjects`, which takes as input a spatial database $\mathcal{D}$ and a set of topological relations $\mathcal{R}$, and returns as output the set of topological objects in $\mathcal{D}$ having a topological relation different from $Disjoint$ and $Universal$ with topological objects described by $\mathcal{R}$; (vi) `computeTopologicalRelations`, which takes as input a set of topological objects (those objects whose identifiers are stored in $\mathcal{A}$), and makes use of method [8] to compute compositions of topological relations among these topological objects.

4 $\mathcal{I}$-SQE: Architecture and Functionalities

$\mathcal{I}$-SQE is characterized by a multi-layer architecture, which is shown in Fig. 2. Each layer of the $\mathcal{I}$-SQE architecture deals with a specific abstraction of the approach we propose for answering range queries over incomplete spatial databases via integrating geometrical information and topological reasoning.

The main components of $\mathcal{I}$-SQE are the following.

***Data Integration Module* (DIM).** This component deals with the problem of integrating spatial data coming from different and heterogeneous spatial information sources, such as conventional spatial databases, spatial data files, and spatial information acquired or inferred from the Web. As highlighted in Sect. 1, integrating spatial data/information poses several issues, such as ensuring the consistency of topological information over the globally-integrated spatial database. The final goal of DIM is that of collecting spatial data from the

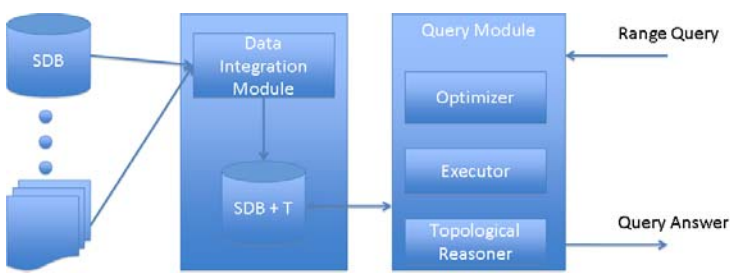

Fig. 2. $\mathcal{I}$-SQE architecture

different sources, and integrating them in a common (spatial) data format while enforcing topological consistency.

***Spatial Database with Topological Information* (SDBT).** In $\mathcal{I}$-SQE, collected and integrated spatial data repositories are materialized within a singleton spatial database where topological information is explicitly represented, named as SDBT. Storing topological information independently of geometrical one allows us to represent in the spatial database even those spatial objects for which the geometry is not known. Another important aspect to be highlighted is about the fact that, in $\mathcal{I}$-SQE, we assume that topological information about spatial database objects is maintained (and indexed) within the spatial database, while new topological relations among the range of the input query and the involved topological objects are computed on-the-fly during query evaluation.

***Query Module* (QM).** Like in a classical query engine, the component *Query Module* of $\mathcal{I}$-SQE embeds a query optimizer and a query executor, for query efficiency purposes (see Fig. 2). In addition to this, $\mathcal{I}$-SQE also embeds the component *Topological Reasoner* that is in charge of inferring topological realtions among topological objects and the range of the input query (see Fig. 3).

Summarizing, given a range query Q over an incomplete spatial database $\mathcal{D}$, $\mathcal{I}$-SQE performs the following steps in order to retrieve the answer to Q, $A(Q)$:

1. Q is parsed by the component *Query Optimizer*, which is in charge of finding an optimal query execution plan for Q, said $\mathcal{P}(Q)$;
2. Q is evaluated against $\mathcal{D}$, and a set of geometrical objects is retrieved from the integrated spatial database SDBT;
3. the component *Topological Reasoner* adds to the set of geometrical objects (retrieved according to the previous point) all the topological objects having a topological relation with the geometrical ones;

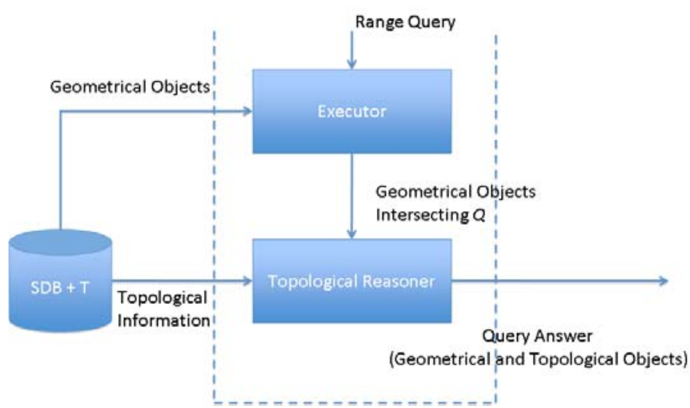

Fig. 3. The component *Topological Reasoner*

4. the answer to Q, $A(Q)$, is retrieved in the vest of a collection of geometrical and topological objects.

5 Conclusions and Future Work

In this paper, we have presented $\mathcal{I}$-SQE, a query engine for answering range queries over incomplete spatial databases via integrating geometrical information and topological reasoning. In particular, we have investigated an application scenario in which topological information exists regardless of geometrical one. We have demonstrated that, in this challenging application scenario, a conventional spatial query engine does not suffice to effectively and efficiently answer range queries, as only geometrical properties of spatial database objects are exploited in order to retrieve the final answers. Contrary to this, $\mathcal{I}$-SQE is able of enhancing the quality and the expressive power of final answers via taking advantages from both the geometrical and the topological representation of spatial database objects, thanks to a nice topological inference approach.

Future work is mainly oriented towards devising solutions for effectively and efficiently answering spatial queries over incomplete spatial databases more complex than simple range queries considered in this research, such as those embedding complex statements like join and selection-partition.

References

1. Open Geospatial Consortium, `http://www.opengeospatial.org`
2. World Wide Web Consortium - SOAP, `http://www.w3.org/TR/soap/`
3. Belussi, A., Bertino, E., Catania, B.: A Reference Framework for Integrating Multiple Representations of Geographical Maps. In: ACM GIS, pp. 33–40 (2003)
4. Belussi, A., Catania, B., Podestà, P.: Towards Topological Consistency and Similarity of Multiresolution Geographical Maps. In: ACM GIS, pp. 220–229 (2005)
5. Butenuth, M., von Gosseln, G., Tiedge, M., Heipke, C., Lipeck, U., Sester, M.: Integration of Heterogeneous Geospatial Data in a Federated Database. International Journal of Photogrammetry and Remote Sensing 62(5), 328–346 (2007)
6. Calì, A., Lembo, D., Rosati, R.: Query Rewriting and Answering under Constraints in Data Integration Systems. In: IJCAI, pp. 16–21 (2003)
7. Dehak, S.M.R., Bloch, I., Maitre, H.: Spatial Reasoning with Incomplete Information on Relative Positioning. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(9), 1473–1484 (2005)
8. Egenhofer, M.J.: Reasoning about Binary Topological Relations. In: SSD, pp. 143–160 (1991)
9. Essid, M., Boucelma, O., Colonna, F.-M., Lassoued, Y.: Query Processing in a Geographic Mediation System. In: ACM GIS, pp. 101–108 (2004)
10. Ives, Z.G., Florescu, D., Friedman, M., Levy, A., Weld, D.S.: An Adaptive Query Execution System for Data Integration. In: ACM SIGMOD, pp. 299–310 (1999)
11. Lin, P.L., Tan, W.H.: An Efficient Method for the Retrieval of Objects by Topological Relations in Spatial Database Systems. Information Processing and Management 39(4), 543–559 (2003)

12. Lin, X., Liu, Q., Yuan, Y., Zhou, X., Lu, H.: Summarizing Level-two Topological Relations in Large Spatial Datasets. ACM Transactions on Database Systems 31(2), 584–630 (2006)
13. Ma, X., Pan, Q., Li, M.: Integration and Share of Spatial Data Based on Web Service. In: IEEE PDCAT, pp. 328–332 (2005)
14. Majkic, Z.: Plausible Query-Answering Inference in Data Integration. In: FLAIRS, pp. 753–758 (2005)
15. Papadias, D., Sellis, T., Theodoridis, Y., Egenhofer, M.J.: Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees. In: ACM SIGMOD, pp. 92–103 (1995)
16. Rodríguez, M.A., Egenhofer, M.J., Blaser, A.D.: Query Pre-processing of Topological Constraints: Comparing a Composition-Based with Neighborhood-Based Approach. In: SSTD, pp. 362–379 (2003)
17. Schockaert, S., Smart, P.D., Abdelmoty, A.I., Jones, C.B.: Mining Topological Relations from the Web. In: IEEE FlexDBIST, pp. 652–656 (2008)
18. Sheeren, D., Mustière, S., Zucker, J.-D.: How to integrate heterogeneous spatial databases in a consistent way? In: Benczúr, A.A., Demetrovics, J., Gottlob, G. (eds.) ADBIS 2004. LNCS, vol. 3255, pp. 364–378. Springer, Heidelberg (2004)