Managing Qualitative Spatial Information to Support Query-by-Sketch

Paolo FOGLIARONI [a,1], Giorgio DE FELICE [a], Falko SCHMID [a], and Jan Oliver WALLGRÜN [b]

[a] *Department of Mathematics and Informatics, Universität Bremen*
[b] *Department of Spatial Information Science and Engineering, University of Maine*

Abstract. In this paper, we propose an extension of existing geographic information systems to support querying-by-sketch. It makes use of a particular class of queries, called qualitative spatial configuration queries, to match sketch-maps against spatial datasets. After analyzing several approaches, we identify the development of a dedicated qualitative layer as the most suitable solution. To face the combinatorial explosion in the number of qualitative relations, we introduce three different strategies to systematically reduce the information to manage. We provide a prototypical implementation for one of them and present empirical results showing the gain in performances when applying the implemented reduction strategy over the case when no reduction is performed.

Keywords. matching, query-by-sketch, qualitative spatial reasoning, volunteered geographic information, qualitative spatial configuration queries

Introduction

VGI [5] is a rapidly developing sector that assumes the involvement of volunteers to collect and provide spatial data to be used in geographic information systems (GIS). Volunteers are, usually, non-experts while GIS, in spite of continuous improvements, mainly provide interfaces tailored for experts. To bridge this gap, new methods and techniques to interact with GIS in a more easy, intuitive and natural way are required. One solution to improve the interaction between user and system can be the utilization of hand-drawn sketch maps to update spatial data or to configure queries on the existing data.

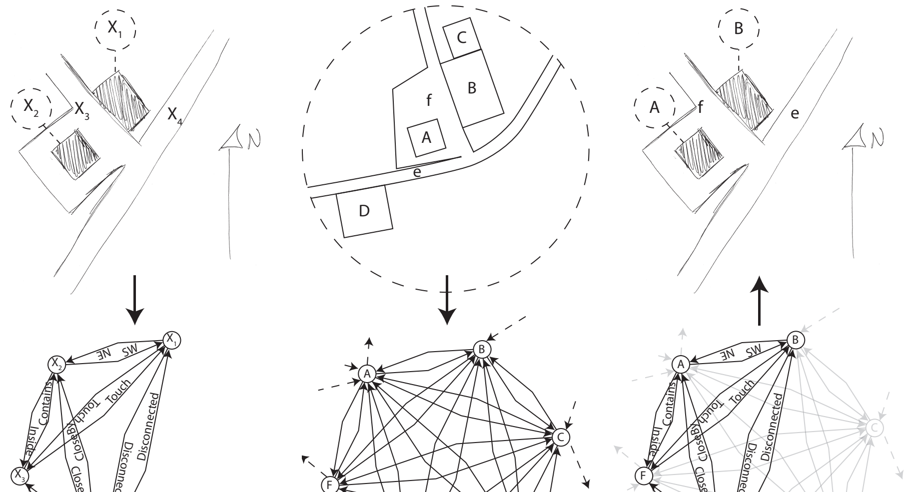
The idea of using sketch-maps to ease the query process ('query-by-sketch') has been around for a while [3]. Sketch-maps clearly have the potential to provide a suitable interface to contribute information to VGI applications. However, in order to employ sketch-maps to interact with GIS, we need methods to match them against spatial datasets. Alignments can be performed geometrically, but due to the imprecise and incomplete nature of sketches they will hardly match exactly with spatial datasets that, on the contrary, provide a precise geometric representation of the real world. An alternative way to perform the matching consists in extrapolating qualitative constraints out of both, the dataset and the sketch, and then performing the alignment at the constraint level [13].

[1]Corresponding Author: Paolo Fogliaroni, Universität Bremen, Department of Mathematics and Informatics, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany; E-mail: paolo@informatik.uni-bremen.de.

As sketch-maps are externalized pictorial representations of human's spatial knowledge, they are usually subject to cognitive distortions introduced in the two-step process of information acquisition (environment learning) and externalization (drawing). However, it has been proved [8,12,14] that certain spatial features are correctly assimilated and reported by human beings. Moreover, cognitive science results [12,7] have shown that humans prefer a qualitative approach to deal with, memorize and reason about space, thus, at least part of the properties preserved in sketch-maps are qualitative in nature.

In the last few decades, a research field dealing with qualitative spatial representation and reasoning (QSR) has developed. QSR techniques abstract from the metric domain into a discrete one (populated by a set of qualitative relations occurring among spatial entities) and provide inference capabilities (see [2] for an overview). Thus, to adequately support applications of VGI that allow for exploiting sketch-maps, the development of tools and solutions for GIS that employ QSR techniques is highly desirable.

Our ongoing work is situated in this context, as we are building a framework that extends GISs' qualitative capabilities. Specifically, we focus on the realization of methods and techniques to perform *qualitative spatial configuration queries* (QSCQ): spatial queries in which one is interested in all sets of objects from the spatial dataset among which a specific spatial configuration is expressed by means of qualitative constraints. This is the kind of queries occurring in query-by-sketch scenarios.



(a) A sketchmap. (b) Part of a spatial dataset. (c) Sketch-dataset matching.

Figure 1. Example: matching a sketchmap against a spatial dataset.

Let us elucidate what a QSCQ looks like and how it can be used for matching purposes with the example depicted in Figure 1. We assume that, either for information retrieval or for contribution purposes (VGI), we need to perform a matching between the sketch in the upper part of Figure 1(a) and a given spatial dataset. Figure 1(b) shows part of the spatial dataset (top) and its corresponding set of qualitative relations (bottom). To perform the match, we need to extract qualitative relations holding among the sketched entities and use them as a QSCQ against the qualitative relations of the dataset. Assuming that the only object configuration in the dataset that matches the relational constraints of the query is the one depicted in Figure 1(b), the query will return exactly this subset of objects together with their mapping to the corresponding objects from the query. As a result, we can now label the entities in the sketch with the matching names (Figure 1(c)).

In the following, we present practical results coming from a prototypical implementation that enables qualitative spatial configuration queries in geographic information systems. The reminder of this text is organized as follows. Section 1 provides a basic introduction to qualitative spatial representation and reasoning and presents previous work on qualitative querying. Practical issues related to the realization of the query processing framework are discussed in Section 2. Section 3 proposes possible solution to the goal of qualitative spatial configuration query enablement, while Section 4 presents some preliminary results of our prototypical framework. Lastly, conclusions are drawn in Section 5, where we also outline next steps towards a full-working framework.

1. Related Work

In the last decades a plethora of qualitative models of space based on finite sets of spatial relations has been proposed as a means to represent human commonsense of space and make it available for computer-based reasoning. However, while qualitative spatial representations and reasoning (QSR) formalisms have been extensively investigated as a subfield of knowledge representation in Artificial Intelligence, solely topological formalisms such as the RCC-8 [10] or 9-intersection model [4] have found their way into existing GIS [6]. There is usually no support for other aspects of space and, in particular, no support to feed qualitative spatial information into the GIS and to query the system using qualitative relations.

The necessity to fill this gap has been recognized in the past. In [3], Egenhofer investigates the possibility to execute spatial queries by expressing spatial constraints through a sketch query language. The same approach is pursued in [11] where a qualitative GIS is presented that allows the user to easily build spatial queries through a graphical drag and drop interface.

The idea of making geographic information systems more suitable to manage qualitative information has also been presented in [15] where the authors introduce the concept of qualitative location, that is a place identified by certain qualitative relations, and develope a prototype implementation to retrieve such locations from spatial datasets.

In [1], Clementini et al. investigate how to deal with the retrieval of topological relations, introducing retrieval strategies based on decision trees. They face the problem of the explosion in the number of qualitative relations between objects in a spatial dataset proving that the retrieval time can be drastically reduced by doing searches in a relation-occurrence-frequency order.

Papadias and Sellis in [9] describe what they call symbolic spatial indexes: arrays preserving only a subset of the spatial relations holding among objects in an image.

Lastly, in [13] possibility to match sketch-maps with quantitative datasets by the combination of graph-subgraph isomorphism and QSR techniques is investigated.

2. Qualitative Spatial Configuration Queries: Feasibility Issues

There exist several ways to answer qualitative spatial configuration queries. In this section we present some of them. We make comparisons and analyze their strengths and weaknesses.

A straightforward approach is to use standard GIS functionalities to develop an abstraction layer that allows to look at geometries in qualitative terms. That is, to implement a set of translational functions to compute at runtime the qualitative relations holding between tuples of geometric objects stored in the GIS. However, beyond the simplicity of its logics, this method turned out to be an unfeasible solution as the computation time of the required functions is linear with the number of spatial relations that, in turn, is quadratic – or cubic when dealing with ternary relations – with the number of spatial entities. Indeed, given a qualitative calculus $\mathcal{Q}\langle a, r\rangle$ with r relations of arity a and a spatial dataset $\mathcal{D} = \{o_1, o_2, \dots o_N\}$ containing N geometric objects, there exists a number of qualitative relations over $\mathcal{D}$ that is equal to the number of *a-tuples* from the dataset:

$$R = \frac{N!}{(N-a)!} \tag{1}$$

To compute such relations, it is necessary to test any *a-tuple* against, in the worst case, every relation in $\mathcal{Q}$ to find out the one that holds. The number of checks K to perform is a function of the calculus and of the dataset at hand and it obeys the following law:

$$K(\mathcal{D}, \mathcal{Q}) = \mathcal{O}\left(r\frac{N!}{(N-a)!}\right) = \mathcal{O}(rR) \tag{2}$$

In [1], it has been shown that the computation time can be drastically reduced by performing the checks according to the relation-occurrence-frequency order. However, even employing this heuristic, computation time becomes quickly intractable for real geospatial datasets.

A further drawback of this solution is that it does not allow to exploit and integrate into the system purely qualitative information, that is, spatial information directly coming in qualitative terms rather than computed from a quantitative dataset—i.e. the kind of information that can be extrapolated from sketch-maps. In our work we do not focus on qualitative information extraction as we assume it is already done. Instead we investigate how to optimize the storage and querying processes.

A different solution consists in extending a GIS with a qualitative storage layer, namely designing an extension of the database schema to explicitly store qualitative spatial relations and a set of procedures to query the qualitatively extended system. This solution overcomes some of the aforementioned problems as it enables the system to deal with a mix of qualitative and geometric information. Moreover, it turns query answering functions into look-up operations, allowing for extremely high performances. However, due to the fact that we still have to compute all of the qualitative relations induced by geometric objects, the computation time remains unfeasible when dealing with real datasets such as the the Bremen OpenStreetMap[2] dataset consisting of nearly 70000 ob-

[2] www.openstreetmap.org/

jects which would mean that to deal with a binary calculus, it is necessary to compute and store a number of relation in the range of 4.8×10^9.

Due to the impossibility to compute all of the qualitative relations in an acceptable amount of time we have to choose a subset of them that we want to store and keep them organized in such a way that retrieval operations stay effective and efficient. Effectiveness is achieved if the totality of the relations can still be obtained—directly or computed—while efficiency can be obtained developing new kind of access methods. The last observations lead us to a third approach that allows us to overcome the relation computation problem. The underlying idea is to compute only a subset of the whole set of qualitative relations (theoretically we do want to compute a number of relations linear with the number of spatial entities) and to infer the rest at runtime. Thus, the extension of a GIS with a qualitative layer together with a relation reduction method and a spatial reasoning module can effectively overcome all of the listed problems, providing a valid framework to answer qualitative spatial configuration queries. However, while this approach reduces the relation computation time, it increases the retrieval time. Thus, it will be necessary to tune the framework parameters (for the reduction strategy and the inference module) to optimize the trade-off between computation and retrieval times.

3. QSCQ Enablement of Geographic Information Systems

In this section, we present different solutions to realize the relation reduction as explained above. We subdivide the relation reduction strategies into three main categories.

Spatial clustering reduction – This approach splits the whole space into zones to only compute relations among objects belonging to the same cluster and between the clusters themselves. The strength of this strategy relies on the fact that, given a dataset $\mathcal{D}$ containing N objects, a calculus $\mathcal{Q}\langle a, r\rangle$ and a space tessellation $\mathcal{T}$ composed of T tiles, we compute a number of relations equal to:

$$R_{\mathcal{T}}(\mathcal{D}, \mathcal{Q}) = T^a + \sum_{i=1}^{T} \frac{n_i!}{(n_i - a)!} \tag{3}$$

where n_i is the number of objects overlapping the *i-th* tile. Opportunely choosing the tessellation technique and strategically tuning its parameters, this value can be much smaller than the cardinality of whole relation set – Equation 1.

QSR-based reduction – This strategy, that is actually under development, is grounded on a novel data structure that we call *inference graph*: a representation of the constraints holding over a spatial dataset that aims at highlighting inference patterns between qualitative relations. An inference graph is an oriented hyper-graph $H = (N, E)$ where N represents the set of qualitative relations holding over the dataset. There exists an oriented hyper-edge $e_{I,j} \in E$ going from a set $I \subset N$ to one node $n_j \in N$ if the relation represented by n_j can be inferred from I. Employing standard graph algorithms, it is possible to decrease the relational node set of an inference graph to a reduced set of relations that still allows to reproduce the dropped ones.

Mixed reduction – This will be a combination of the first two methods that takes advantage of the properties of both of them. The two strategies either can be applied consecutively or they can be somehow fused together. In the first case, we can first split the space and then apply the QSR-based reduction on every tile, while for the second combination approach we can device another level of reasoning that exploits interconnections between inference graphs belonging to different tiles to further refine the reduction.

In the ongoing work we research, implement and test solutions to realize the three categories above with respective reasoner modules. We evaluate the efficiency of every solution while varying geometric dataset characteristics to detect optimal values for every kind of scenario. The characteristics we focus on are relation computation performances and QSCQ response time. We expect each method to perform better with specific classes of spatial datasets and QSCQs. In this work, we only focus on the spatial clustering reduction and present practical results for a possible implementation.

4. Preliminary Results: A *Grid*-based Approach

To evaluate our approach, we implemented a relation reduction strategy based on space tessellation by means of a *grid*. Although there exist more sophisticated indexing strategies —i.e. R-trees, quad-trees, etc.— for this work we decided to opt for a *grid* tessellation because it provides a straightforward and powerful indexing mechanism. We resort to a *grid* composed of uniform square cells covering the selected spatial dataset. In this case, both variables n_i and T in Equation (3) depend on the *grid* cell dimension C that is effectively the only parameter of interest when applying a *grid* tessellation.

As discussed in Section 2, when applying no reduction strategies, the main issue relies in the time required to compute the relations. For this work we only consider the RCC-8 calculus [10] that defines eight binary topological relations between regions in a topological space. Hence, according to Equation (1), we have $R = \frac{N!}{(N-2)!} = N(N-1)$.

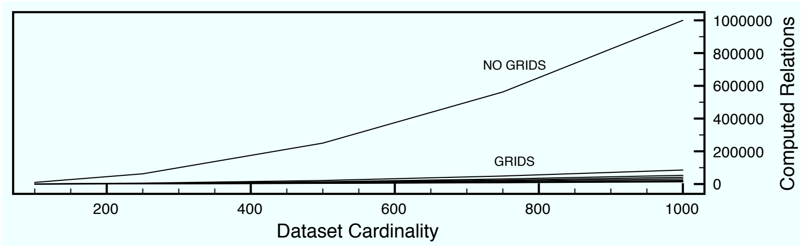
On the other hand, when applying a relation reduction strategy, we expect a reduction in the relation computation time at the cost of an increased overhead in the retrieval phase, as we will have to infer relations not explicitly stored. However, as will be explained in section 4.2, for RCC-8 it turned out that, a *grid* reduction improves retrieval performances rather than worsen them.

We developed a prototypical implementation that extends an existing spatial-extended DBMS and runs tests over 100 datasets. Each dataset consists of randomly generated, polygons uniformly distributed in a $D \times D = 500 \times 500$ fixed-size workspace. The number of polygons N and the average object dimension $\underline{d}$ have been ranged over the sets $\{100, 250, 500, 750, 1000\}$ and $\{7.5, 15\}$, respectively. Specifically we generated 10 datasets for every pair $(N, \underline{d}) \in \{\{100, 250, 500, 750, 1000\} \times \{7.5, 15\}\}$. Experiments have been run in two main stages: relation computation and QSCQ executions.

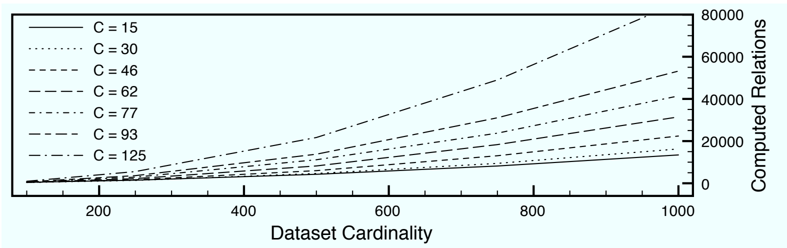
4.1. Relation Computation

Every dataset has been tested against 8 configurations—7 with different *grid* parameter values C and one with no applied reduction strategy, this means without grid tessellation or another indexing technique—for a grand total of 800 relation computation tests.

Our results (see Fig. 2) show that the number of relations R_{grid} to compute is much smaller than the size R of the whole relation set, we compute when applying no reduc-



(a) Number of relations computed with and without *grid* ($\underline{d} = 15$).



(b) Section from Figure 2(a) focusing on the different grid parameters C.

Figure 2. Number of computed relations as the dataset cardinality varies.

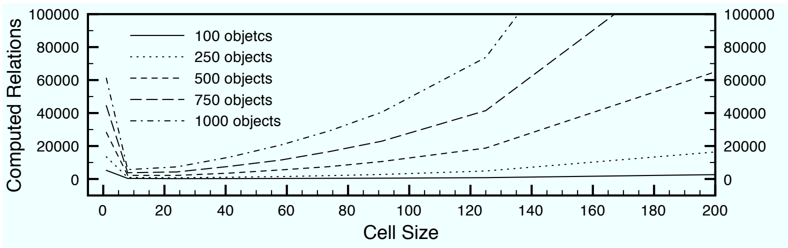
tion strategies. Figure 2(a) shows a set of curves representing the number of cimputed relations as the dataset cardinality N varies with and without the *grid*-based reduction strategy. The *grid* approach performs many orders of magnitude better independent of the the particular parameter C chosen.

Figure 2(b) focuses on the *grid* performances for different values of C. It is evident that the smaller the cell dimension, the better the performance. However, if cells are smaller than the average object size, an object will, on average, belong to multiple cells which implies an increase in the number of objects per cell and, thus, of the computed relations. On the other hand if cells are greater than or equal to the whole workspace, one cell would contain all of the objects—that is like applying no reduction strategy.

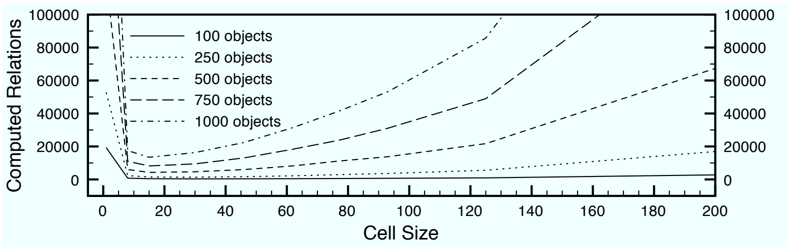
Although the grid tessellation is a widely investigated approach we decided to run additional tests to empirically check what is the optimal value for C in terms of computed relations. The expectation is that by reducing C, the number or relations grows down to a minimal value occurring when C reaches the optimal value $\hat{C}$, whereas reducing C beyond this limit causes R_{grid} to increase again. Such expectations have been indeed confirmed by our experiments as shown in Figures 3(a) and 3(b). The main difference between the two graphics lies in growing rate of the number of computed relations when $C < \hat{C}$. Indeed, the curves for datasets with $\underline{d} = 15$ grow much faster because the objects are in average bigger than those in the other datasets, thus they occupy more cells inducing a bigger number of relations. Further experiments have shown $\hat{C}$ to be approximately equal to the average object dimension $\underline{d}$. Thus, given a geospatial dataset, it is possible compute the optimal *grid* in linear time by a single scan of the dataset.

4.2. Qualitative Spatial Configuration Queries

Qualitative relations generated in the previous stage have been explicitly stored in dedicated tables that in the remainder of the text will be referred with the name *relation*



(a) Average object dimension $\underline{d} = 7.5$



(b) Average object dimension $\underline{d} = 15$

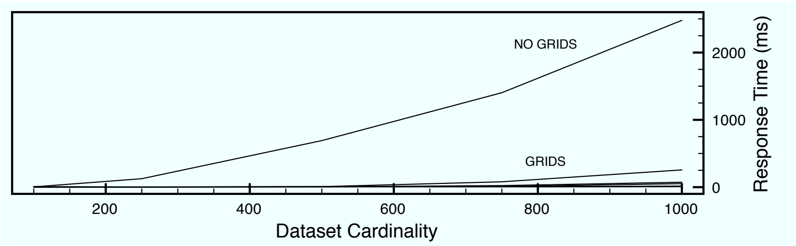
Figure 3. Details on the number of computed relations as the cell dimension varies.

*table*s. In this stage, we ran QSCQs on all of the 800 configuration coming from the relation generation phase. We executed 11 different QSCQs varying the cardinality of the requested configuration and the topological constraints in the queries. Eight of them request configurations of two objects among which a particular RCC-8 relation holds. The others require three objects with different relations holding among them.

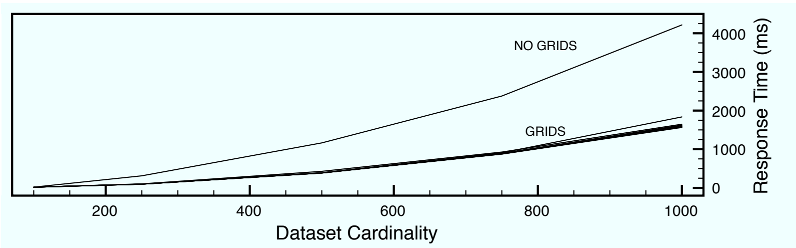
For settings with no reduction strategies, the retrieval phase simply consists in a standard query over the corresponding *relation table* that, indeed, contains all of the actual relations.

Things are just slightly harder when dealing with *grid*s. When looking for qualitative relations other than DC, the retrieval strategy stays as simple as in the previous case, as we can directly retrieve them from the *relation table*. What differs is the retrieval strategy for DC relations as, in the relation generation phase, we computed and stored only a part of them. To rebuild the missing relations we would need to perform a series of inference steps that would worsen the retrieval performances. However, when applying the *grid* reduction strategy and dealing with RCC-8 relations, we can avoid to go through this inference process and easily rebuild them all by means of a set difference operation: subtracting from the set of all pairs of objects in the dataset the set of object pairs among which a non-DC relation holds. Such a strategy cannot not be the applied in general case, thus, for other calculi we expect the retrieval time to be higher when using reduction strategies.

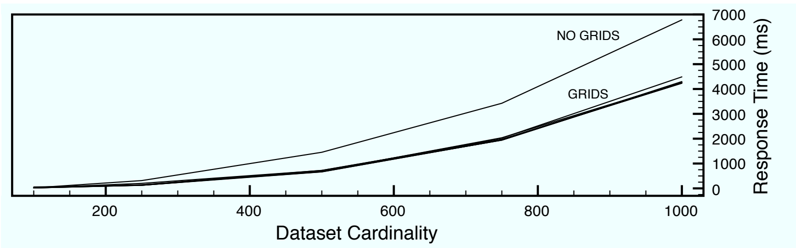
Figure 4 shows curves for the retrieval times for different QSCQs as the number of objects increases for the case with no reduction strategy and for different values of C in the case of *grid* reduction. In every case, *grid*-reduced instances perform better than unreduced ones. Particularly, as we can see in Figure 4(a), the retrieval time for PO relations is extremely fast when using grids. The performances remain the same for every relation but DC which, as shown in Figure 4(b), reaches retrieval times much closer to



(a) $QSCQ = PO(X_1, X_2)$.



(b) $QSCQ = DC(X_1, X_2)$.



(c) $QSCQ = PO(X_1, X_2) \wedge PO(X_2, X_3) \wedge DC(X_1, X_3)$.

Figure 4. QSCQ response times with and without *grid* as the dataset cardinality varies ($\underline{d} = 15$).

the case with no reduction strategy. Furthermore, Figure 4(c) reveals that DC retrieval times seem to dominate other constraints and thus to degrade overall performances when appearing in a QSCQ.

5. Conclusions and Future Work

In this paper we suggested several ways to enable qualitative spatial configuration queries in spatial databases to support querying-by-sketch, discussing singular weaknesses and strengths. We came to the conclusion that one feasible solution consists in supplying present systems with a qualitative storage layer, to store and query qualitative spatial data. Furthermore we have shown that to keep relation computation times and QSCQ retrieval times feasible for real datasets, it is necessary to store only a selected subset of the whole relation set. We suggested three different relation reduction strategies—space clustering, QSR-based, and a mixed approach—to systematically reduce the relation space and provided empirical results for a *grid*-based spatial clustering solution. The three methods will be further investigated empirically in the future including an analysis of the reduction properties of the most important spatial access methods, R-trees, quad-trees, etc. For the QSR-based approach, we will develop a novel data structure named *inference graph* that encodes inference properties of spatial calculi and allow, through the use of standard graph algorithms, to obtain a reduced set of relations. Again, we plan to integrate the first two reduction strategies in one unique methodology to combine their individual strengths. Lastly, experiments of the same kind of those documented in this text will be performed on every solution.

Acknowledgements

The authors would like to thank Christian Freksa and the anonymous reviewers for valuable comments. Funding by the Deutsche Forschungsgemeinschaft (DFG) under grant IRTG Semantic Integration of Geospatial Information is gratefully acknowledged.

References

[1] E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling topological spatial relations: Strategies for query processing. *Computers & graphics*, 18(6):815–822, 1994.

[2] A. G. Cohn and S. M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29, 2001.

[3] M. J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, 1997.

[4] M. J. Egenhofer and R. D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Science*, 5(2):161–174, 1991.

[5] M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[6] J. Herring. The opengis abstract specification, topic 1: Feature geometry (iso 19107 spatial schema), version 5. *OGC Document*, pages 01–101, 2001.

[7] B. Kuipers. Modeling spatial knowledge. *Cognitive science*, 2(2):129–153, 1978.

[8] K. Lynch. *The image of the city*. the MIT Press, 1960.

[9] D. Papadias and T. Sellis. Qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal*, 3(4):479–516, 1994.

[10] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann, 1992.

[11] C. P. L. Schultz, H. W. Guesgen, and R. Amor. Computer-human interaction issues when integrating qualitative spatial reasoning into geographic information systems. In *Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI*, pages 43–51. ACM, 2006.

[12] B. Tversky. Distortions in memory for maps. *Cognitive Psychology*, 13(3):407–433, 1981.

[13] J. O. Wallgrün, D. Wolter, and K. F. Richter. Qualitative matching of spatial information. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 300–309. ACM, 2010.

[14] J. Wang and A. Schwering. The accuracy of sketched spatial relations: How cognitive errors affect sketch representation. In Thora Tenbrink and Stephan Winter, editors, *Presenting Spatial Information: Granularity, Relevance, and Integration, 2009. International Workshop in conjunction with COSIT 2009.*, 2009.

[15] X. Yao and J. C. Thill. Spatial queries with qualitative locations in spatial information systems. *Computers, environment and urban systems*, 30(4):485–502, 2006.