POINT-SET TOPOLOGICAL RELATIONS PROCESSING IN IMAGE DATABASES *

John Z. Li and M. Tamer Özsu
Department of Computing Science, University of Alberta, Canada

Abstract

Egenhofer and Franzosa's model of fundamental topological relations for spatial regions has received a lot of research attention in geographic information systems and spatial databases. We propose a new way of computing these topological relations with much less storage requirement. We investigate different cases where the new approach can perform even better than the original approach in terms of CPU time. All the experiments are run on top of a commercial object database management system. Some important factors which impact the performance of computing topological relations are also discussed in detail. An image database prototype has been built based on this new approach.

KEYWORDS: spatial region, multimedia, topological relation, point-set.

1 INTRODUCTION

Management of multimedia data poses special requirements for database management systems (DBMSs). Many applications depend on spatial relationships among multimedia data. There is significant research on spatial relationships in geographic information systems (GIS) and image databases [1, 2, 3, 4, 5, 6]. Part of this research is on supporting content-based retrieval, which is the most striking difference between multimedia DBMSs and their traditional counterparts. Spatial relations have been classified [7] into several categories, including *topological relations* that describe neighborhood and incidence (e.g., overlap, disjoint); *directional relations* that describe order in space (e.g., south, northwest); and *distance relations* that describe space range between objects (e.g., far, near).

Users are always interested in the topological relations between regions – intuitively, those that pertain to connectivity of the regions. For example, "*Find all satellite images which show two adjacent cities or towns each with population more than 10000*". In many cases the precise knowledge of the regions is important. The point-set approach [1, 8] is the most general model for the representation of spatial regions. However, point-set approaches are notorious for large storage space requirement. Saving space with a small degradation in computation efficiency is the major motivation of our work. Directly extracting point-sets from raw images is too expensive and almost impossible in practice. As some research [6] indicates we can reduced the size of raw images, say to 64×64, without losing their spatial relations.

In this paper, we propose a new way of computing point-set topological spatial relations by eliminating the *interior* point-set. This will not only dramatically reduce the database size, but also speed up query evaluation in some cases. The major contributions of this work are: a completely new way of computing point-set topological spatial relations; experimental data to show the efficiency and feasibility of the new approach; and an image database prototype

*This research is supported by a strategic grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

available for public accesses. We are using an object database management system, ObjectStore [9], as spatial object repository.

2 RELATED WORK

Egenhofer and Franzosa [1] have specified eight fundamental topological relations that can hold between two planar regions. These relations are computed using four intersections over the concepts of *boundary* and *interior* of point-sets between two regions embedded in a two-dimensional space. For example, let A° and B° be the interiors of objects A and B respectively and $A^\bullet$ and $B^\bullet$ be the boundaries of A and B respectively, then the combinations of intersections ($A^\circ \cap B^\circ$, $A^\circ \cap B^\bullet$, $A^\bullet \cap B^\circ$, $A^\bullet \cap B^\bullet$) between interiors and boundaries form a set of topological relations. By checking each intersection is *empty* or *not empty*, there are 16 possible results. However, only eight meaningful topological relations are identified: *disjoint, contains, inside, touch, equal, covers, covered_by*, and *overlap* as shown in Figure 1.

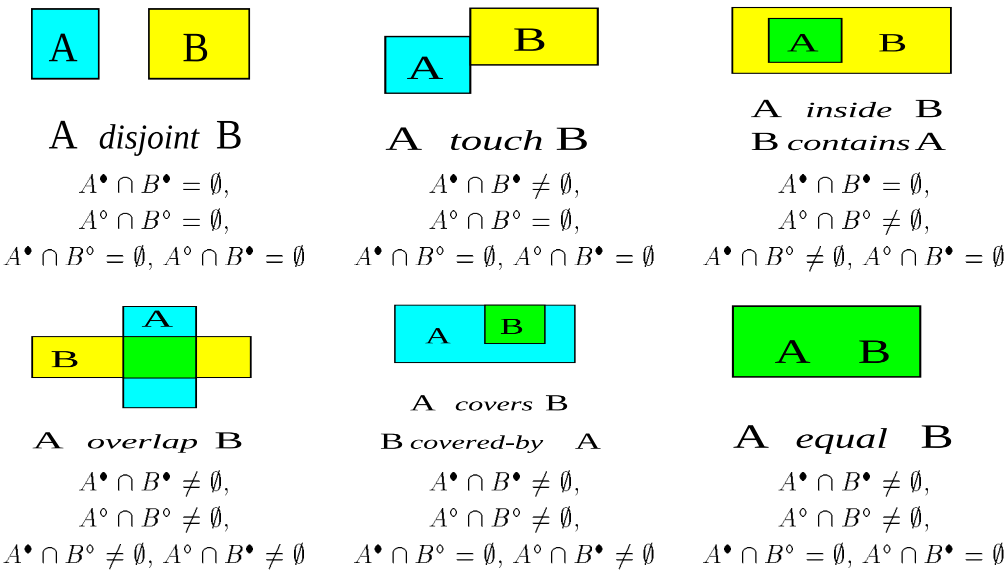


Figure 1: Definitions of Topological Relations

Papadias et al. [3, 4] assume a construction process that detects a set of special points, called *representative points*, in an image. Every spatial relation in the modeling space can be defined using only the representative points. Two kinds of representative points are considered: *directional representative points* and *topological representative points*. Their topological reasoning work is based on Egenhofer and Franzosa's eight topological relations in two dimensional space. The topological relations are divided into three levels of resolution (high, medium, and low) according to the applications. The objective is to reduce the computational complexity whenever possible by a using lower level resolution.

Based on point-set approach, a sound and complete spatial reasoning system is presented in [8]. The soundness and completeness require that each object be *connected*, which means that the object does not have disjoint parts. The reasoning model is expressed in rules which can be easily integrated into a spatial database. However, a serious drawback of this inference system is its low expressive power. There are only four directional relations and three topological relations in two dimensional space, which is not enough.

3 EXPERIMENT

The model of topological spatial relations is based on the point-set topological notions of *interior* and *boundary*. Following theorem tells us that the interior point-set can be eliminated in computing topological relations. The proof is omitted because of the space limit.

Theorem Let A be a convex region, A° be A's interior and $A^\bullet$ be A's exterior point-sets respectively. For all $p \in A^\circ$ there always exist four points p_1, p_2, p_3, p_4 in $A^\bullet$, i.e., $p_i \in A^\bullet$ $(i = 1, 2, 3, 4)$, such that

$\mathcal{F}_x(p_1) = \mathcal{F}_x(p)$ and $\mathcal{F}_y(p_1) < \mathcal{F}_y(p)$, $\mathcal{F}_x(p_2) = \mathcal{F}_x(p)$ and $\mathcal{F}_y(p_2) > \mathcal{F}_y(p)$,

$\mathcal{F}_y(p_3) = \mathcal{F}_y(p)$ and $\mathcal{F}_x(p_3) < \mathcal{F}_x(p)$, $\mathcal{F}_y(p_4) = \mathcal{F}_y(p)$ and $\mathcal{F}_x(p_4) > \mathcal{F}_x(p)$.

where $\mathcal{F}_x$ and $\mathcal{F}_y$ map a point into the projections of an x-axis and y-axis respectively.

In order to see the performance of this approach for computing topological relations, we have designed and run following experiments over ObjectStore using C++. The goal of this experiment is to evaluate the performance of two different approaches on computing point-set topological relations. The first approach is to explicitly store the interior of a spatial region and the second approach is to replace such an interior with a function. Let us call the first approach *original approach* and the second approach *functional approach*. The performance metrics are CPU time required in computing topological relations.

The experiment design technique being used is the 2^k factorial design [10]. 2^k factorial design is an effective experiment method for evaluating effects of some factors and their combinations. k represents the number of factors and 2 refers to the levels of each factor. We choose three factors in our experiment; therefore, we use 2^3 factorial design. The three factors are: (1) Average sizes of interiors of spatial regions; (2) Average sizes of boundaries of spatial regions; (3) Whether point-sets have index support or not. Since our argument is to eliminate spatial region's interior explicitly, it is natural to see how different size relationships between interiors and boundaries of spatial regions influence the overall performance, i.e., the CPU time. It is a common technique to use indexes in databases to handle large amount of data. We would like to see how indexes affect the overall performance too. One thing to point out is that in the case of using indexes we do not consider the time to create and maintain the indexes. The cost of maintaining indexes can be very high if there are many region updates. The two levels for average sizes of region interiors and region boundaries are both 100 points and 200 points respectively. We use two-dimensional hash index over a point. The maximum testing size is 200 spatial regions which are randomly generated. The final database size is at around 8M bytes and all the experiments are conducted on a SUN SPARC SS20 with a 70MHZ cpu and a 64MB memory.

Table 1 and Table 2 show the results of our experiments over the three factors: interior size (Interior-100 and Interior-200), boundary size (Boundary-100 and Boundary-200), and indexable (Hash or no Hash). For each spatial region, a pair-wise comparison over all other spatial regions is computed for each of the eight topological relations. The average number of of topological relation comparisons are 135,000. In general, the original approach has a better performance than functional approach when the testing size is small.

Index	Interior-100		Interior-200	
	Boundary-100	Boundary-200	Boundary-100	Boundary-200
No Hash	489	257	205	194
Hash	201	163	107	108

Table 1: CPU Times (seconds) of the Original Approach

Index	Interior-100		Interior-200	
	Boundary-100	Boundary-200	Boundary-100	Boundary-200
No Hash	471	288	242	218
Hash	266	240	134	194

Table 2: CPU Times (seconds) of the Functional Approach

However, the purpose of this 2^3 factorial experiment is not to compare the performance between these two approaches; it is to see the impact of each factor and their combinations over each approach. Using the method, discussed in [10], of computing the effect of each factor, we come out Table 3 from Table 1 and Table 2. In Table 3 the *Interior* is the effect of interior sizes of spatial regions; the *Boundary* is the effect of boundary sizes of spatial regions; and the *Hash* is the effect of hash index. "I-B" is the effect of the combination of *Interior* and *Boundary*; "I-H" is the effect of the combination of *Interior* and *Hash*; "B-H" is the effect of the combination of *Boundary* and *Hash*; and "I-B-H" is the effect of the combination of *Interior*, *Boundary*, *Hash*.

In the case of the original approach, indexing has played an important role with 39% contribution to the total CPU time while the sizes of region boundaries contribute 30% which is also significant. This reveals that region boundaries are more important than region interiors because it seems they consume more CPU time in the computation of topological relations. The combination of *Interior* and *Hash* contributes 8% and other combinations have little impact on the overall performance.

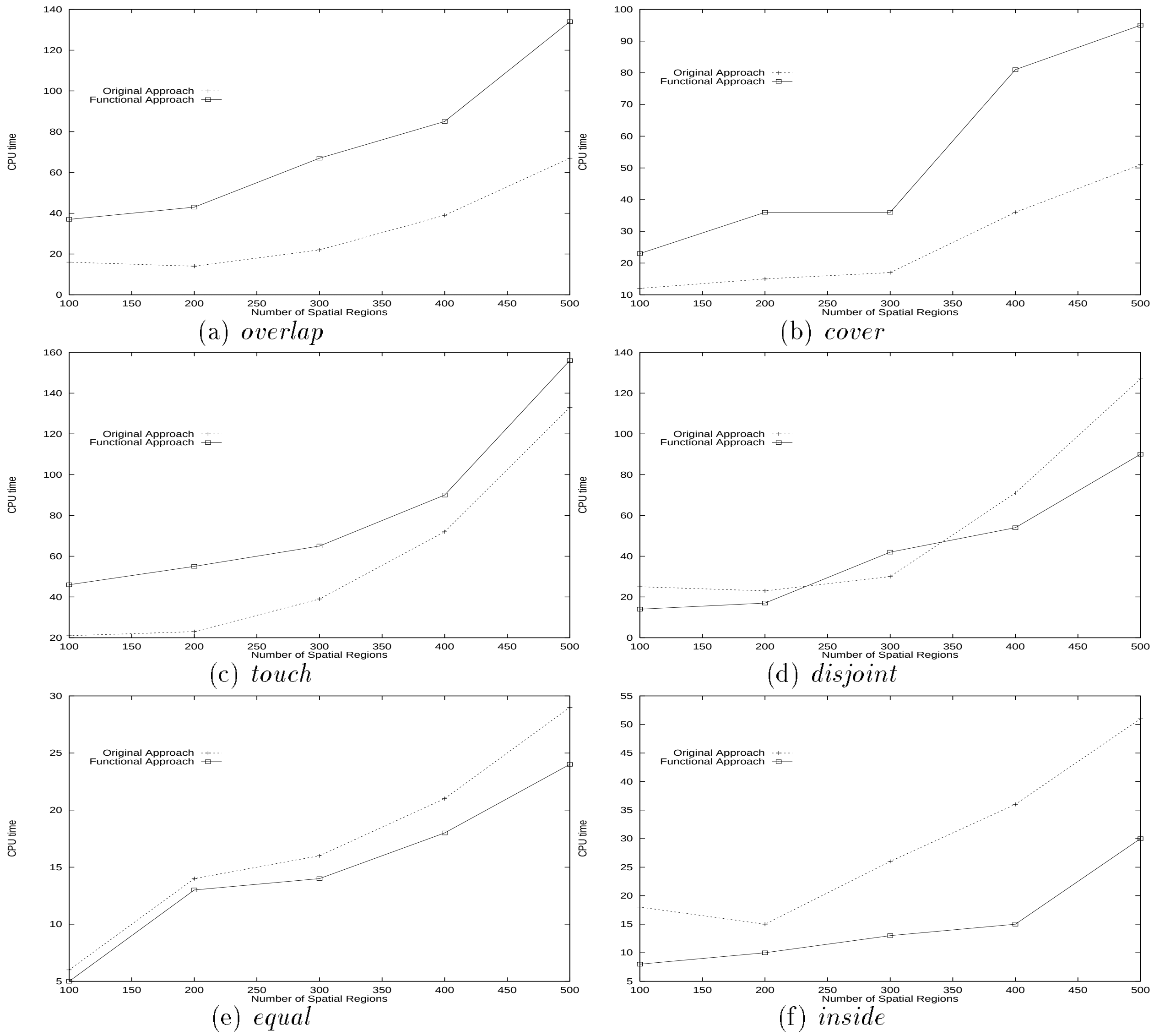
In the case of the functional approach, indexing has played a less important role (28% of total CPU time) compared with the original approach (39%). This makes sense because in the functional approach there are no interiors for spatial regions so indexes are only for the boundaries. On the other hand, boundaries of spatial regions (42%) have bigger impact compared with the case in the original approach (30%). The reason is that we are relying on region boundaries more than before. The interiors contribute only 5% of the total performance. The combination of region interiors and boundaries have 11% effect which is quite significant. So does the combination of region interiors and hash index. The contribution of the combination of all the three factors is negligible.

Approach	*Interior*	*Boundary*	*Hash*	I-B	I-H	B-H	I-B-H
Original	10%	30%	39%	8%	5%	5%	3%
Functional	5%	42%	28%	11%	11%	2%	1%

Table 3: Percentage of Effects of 3 Factors over Different Approaches

Figure 2 shows different CPU times in computing different topological relations. In Figure 2 (a) (*overlap*), (b) (*cover*), and (c) (*touch*) the original approach beats the functional approach with quite a bit of margin while in Figure 2 (d) (*disjoint*) the two approaches are alternately taking the leads. This reveals that when there are many comparisons to region interiors (as the cases in computing *overlap* and *cover* relations) the original approach performs better. This is because region interiors have been indexed in the original approach while there is no index at all in the functional approach.

In Figure 2 (e) (*equal*) and (f) (*inside*), the functional approach performs consistently better than the original approach. The reason is that the computation is mainly dependent on region



(a) *overlap* (b) *cover*
(c) *touch* (d) *disjoint*
(e) *equal* (f) *inside*

Figure 2: Different Topological Relations (CPU time in seconds)

boundaries so the computation workload is roughly the same with both two approaches. For the functional approach the database is much smaller so it requires less CPU and I/O time to do the processing. That is why the functional approach out-performs the original approach in these two cases. In summary the functional approach has better performance than the original approach in some cases. These graphs also reveal that the most expensive topological relations in terms of CPU time are *touch*, *overlap*, and *disjoint*.

4 CONCLUSIONS

Different applications of spatial databases pose different requirements on data representations. Although higher level of abstraction over spatial objects are desirable, sometimes it is not possible to attain these abstractions. This is true especially in the case of multimedia system, which could be either because image processing techniques are not good enough or the quality of multimedia objects (e.g., images) is not high. We see the point-set approach as the most general model for the representation of spatial regions.

In this paper, we propose a new way of computing point-set topological spatial relations by eliminating the *interior* point-set, which dramatically reduces the database size and speeds up query evaluation in some cases. Although in terms of CPU time performance of the functional approach is not as good as that of the original approach, our work does show that it is a good strategy to use the functional approach if a limited increase in computation time is acceptable in order to save space. As a matter of fact, the major issue concerning spatial and multimedia databases is the huge volume of data. Storage space is, and will still be, a big problem. Although our approach applies only to the convex regions, it may also be applied to non-convex regions because a non-convex region can be divided into a set of convex regions.

We have developed an image database using the functional approach. A two-level approach is used to process spatial queries. At first level (or filter level), all the spatial regions are approximated by their minimum bounding rectangles (MBRs). The computation of topological relations between MBRs are very simple [5]. Possible candidates generated from this level are passed to the next level which is the point-set model. Hopefully this level deals with significantly less spatial regions than the first level. This image database supports a wide range set of spatial queries. One goal, which we achieved, is to allow this system being queried by either traditional way (key-word) or the content-based retrieval approach. Furthermore, the combination of these two is also possible. Our image database contains around 500 images and a demo version is publically accessible from `http://www.cs.ualberta.ca/~zhong` through any popular web browser (e.g., Netscape).

References

[1] M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *Int'l Journal of Geographical Information Systems*, 5(2):161—174, 1991.

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

[3] D. Papadias and T. Sellis. The qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal*, 4:100—138, 1994.

[4] D. Papadias, Y. Theodoridis, T. Sellis, and M. J. Egenhofer. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *Proc. of ACM SIGMOD Int'l Conf. on Management of Data*, pages 92—103, San Jose, CA, May 1995.

[5] J. Z. Li, M. T. Özsu, and D. Szafron. Modeling of video spatial relationships in an object database management system. In *Proc. of the Int'l Workshop on Multimedia Database Management Systems*, pages 124—133, Blue Mountain Lake, NY, August 1996.

[6] D. A. White and R. Jain. ImageGREP: Fast visual pattern matching in image databases. In *Proc. of Storage and Retrieval for Images and Video Databases V, IS&T/SPIE Symp. on Electronic Imaging Science and Technology*, pages 96—107, San Jose, CA, 1997.

[7] D. Pullar and M. Egenhofer. Toward formal definitions of topological relations among spatial objects. In *Proc. of the 3rd Int'l Symp. on Spatial Data Handling*, pages 165—176, Sydney, Australia, 1988.

[8] P. Sistla, C. Yu, and R. Haddack. Reasoning about spatial relationships in picture retrieval systems. In *Proc. of the 20th Int'l Conf. on VLDB*, pages 570—581, Chile, 1994.

[9] C. Lamb, G. Landis, J. Orenstein, and D. Weinreb. The ObjectStore database system. *Communications of ACM*, 34(10):19—20, 1991.

[10] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.