

Consistent Queries over Cardinal Directions across Different Levels of Detail*

Roop K. Goyal and Max J. Egenhofer
National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
348 Boardman Hall
University of Maine
Orono, ME 04469-5711
{rgoyal, max}@spatial.maine.edu

Abstract

Current models for cardinal directions, such as *north* and *northeast*, are either point-based or region-based, but no models exist that apply equally, independent of the geometric data types (be it points, lines, or polygons). To allow users to formulate queries such as “Find all towns in Maine that are *northeast* of Augusta” without pondering about the cities’ geometric data types, we extend the model of the model of the *direction-relation matrix* to handle arbitrary pairs of points, lines, and polygons. This new model, called the *deep direction-relation matrix*, retains the 3×3 structure of the direction-relation matrix with empty and non-empty tiles, while it records additionally neighbor codes for empty tiles to capture whether the tiles’ boundaries are empty or not. This extension covers all intricacies imposed by line and point objects, yielding a unifying and consistent model for cardinal directions. It enables the use of cardinal directions in spatial query languages independent of the objects’ geometric data types.

1. Introduction

Spatial database systems used for geographic information systems (GISs) support spatial queries based on the objects’ geometries and non-spatial attributes [5]. They typically encode the geometry of spatial objects in terms of points, lines, and polygons. Depending on the level of detail, the same spatial object may be recorded as a polygon or a point (e.g., for cities) or as a polygon or a line (e.g., for roads). When using spatial data in a GIS, users have to know about the geometric data types in order to apply appropriate operations, because the semantics and definitions of the spatial relations may differ depending on the representation of the objects. The overloading of operations and polymorphism [4] have found attention in GIS, for instance with Smallworld’s macro-language Magik [13], and spatial database systems [21]. While such methods would help in minimizing the number of operations at an interface and, thereby, reducing the cognitive load currently imposed upon users of GISs and spatial database systems, none of today’s models for qualitative directions is defined at a level sufficiently independent of the objects’ representations. The advent of multi-representation geographic databases [2, 3, 18] will only aggravate the problem of querying across different geometries.

This paper considers direction relations between two objects at different levels of detail, where generalized spatial objects may change their dimensional representation from a polygon to a line, or from a polygon to a point [11]. Cardinal directions, such as *north* and *northeast*, are frequently used in spatial queries as constraints to retrieve spatial configurations [15, 20, 22]. They capture spatial properties that are variant under rotations. Unlike quantitative directions, which refer to detailed numerical values in degrees, minutes, and seconds, cardinal directions are qualitative spatial relations and, therefore, are based on a small set of symbols [7, 10]. Most existing models of cardinal directions use either a cone-shaped [17] or projection-based [6] reference frame, which limits their use to point representations. For polygons or lines these models require each object to be represented by a single point, such as the object’s geometric center. Other models of cardinal

* This work was partially supported by the National Imagery and Mapping Agency under grant number NMA202-97-1-1023. Max Egenhofer’s research is further supported by the National Science Foundation under grant numbers 9613646, SBR-9600465, BDI-9723873, EIA-9876707, and IIS-9970123; the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01.

directions approximate objects by their minimum bounding rectangles (MBRs) [15], which often leads to erroneous query results. The *direction-relation matrix* [9] was introduced to capture cardinal directions between areal objects without applying an approximation. While the model achieves this goal, it lacks compatibility with point-like and certain linear spatial objects. The goal of this paper is to define a unifying knowledge structure that (1) distinguishes direction relations for points, lines, and polygons and (2) groups direction relations over dimension-varying objects into compatible categories. The advantage of such an integrated direction model is that it applies consistently across different spatial data types—points, lines, and polygons—so that users need not be aware of the geometric data types of the queried objects.

The remainder of this paper is structured as follows: Section 2 reviews the direction-relation matrix for polygons. Section 3 analyses the shortcomings of the direction-relation matrix for lines or points, and Section 4 introduces the deep direction-relation matrix, which overcomes these limitations without additional cost for polygon-polygon relations. Section 5 analyses the direction-based queries across different levels of detail. Conclusions follow in Section 6.

2. Cardinal Directions between Two Polygons

A cardinal direction is a triple $\langle A, d, B \rangle$, where A and B are the *reference object* and *target object*, respectively, and d is a non-empty subset of the nine symbols $\{N, S, E, W, NE, SE, SW, NW, O\}$, whose semantics are motivated by the compass rose. Goyal and Egenhofer [9] introduced a direction-relation model for extended spatial objects that considers the influence of the objects' shapes. It uses the projection-based direction partitions [6] and an extrinsic reference system [19], and considers the exact representation of the target object with respect to the reference frame. The reference frame with a polygon as reference object has nine direction tiles: north (N_A), northeast (NE_A), east (E_A), southeast (SE_A), south (S_A), southwest (SW_A), west (W_A), northwest (NW_A), and same (O_A). The cardinal direction from the reference object to a target is described by recording those tiles into which at least one part of the target object falls (Figure 1).

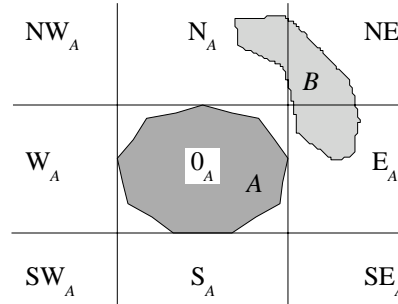


Figure 1: Capturing the cardinal direction relation between two polygons, A and B , through the projection-based partitions around A as the reference object.

For cardinal directions between two polygons, a 3×3 matrix captures the neighborhood of the partition around the reference object and registers the intersections between the target and the tiles around the reference object (Equation 1). The elements in the direction-relation matrix have the same topological organization as the partitions around the reference object.

$$dir_{RR}(A, B) = \begin{bmatrix} NW_A \cap B & N_A \cap B & NE_A \cap B \\ W_A \cap B & O_A \cap B & E_A \cap B \\ SW_A \cap B & S_A \cap B & SE_A \cap B \end{bmatrix} \quad (1)$$

To describe coarse cardinal directions, we consider the emptiness and non-emptiness of the nine intersections between the nine tiles formed around the reference object and the exact representation of the target object. Equation 2 shows the direction-relation matrix for the configuration in Figure 1.

$$dir_{RR}(A, B) = \begin{bmatrix} \emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \quad (2)$$

3. Applying the Direction-Relation Matrix to Lines and Points

Since the representation of a spatial object may change across different levels of spatial detail, it is necessary for a comprehensive direction-relation model to be consistently applicable across different dimensional representations of the objects. Two issues need to be considered: (1) the influence of the reference object's dimension and (2) the variations that arise due to different dimensional types of target objects. This section illuminates the implications of using the vanilla direction-relation matrix model for points, lines, and polygons.

3.1 Dimension Variations of the Reference Object

For the direction-relation matrix, a reference grid is based on an orientation and the extent of the reference object. To illustrate the issue, we keep the orientation fixed, because this is the fundamental property of an extrinsic reference frame, but modify the type of the reference object. At different levels of detail, the reference object may be a polygon, a line, or a point. Since the direction-relation matrix forms the central tile around the reference object, the choice of a polygon, a line, or a point influences the construction of the reference grid. The different configurations can be derived as follows: The projection of an object onto a grid axis results either in a line or a point. For a reference grid with a vertical and a horizontal axis, there are four possible combinations that yield different types of reference frames (Figure 2): (1) both projections are points; (2) a point on the horizontal axis and a line on the vertical axis; (3) a line on the horizontal axis and a point on the vertical axis; and (4) both projections are lines. The first three cases describe unique configurations when the object is a point, a vertical line, and a horizontal line, respectively. Case 4 occurs when the reference object is a polygon or a line that is neither strictly horizontal nor vertical.

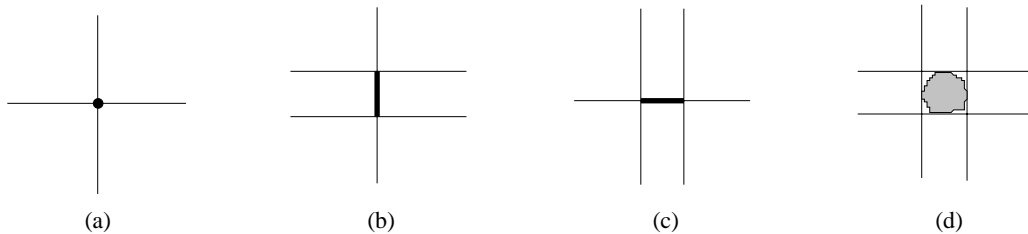


Figure 2: Variations for reference frames based on different types of objects: (a) for a point, (b) for a vertical line, (c) for a horizontal line, and (d) for other lines or all polygons.

3.2 Dimension Variations of the Target Object

In a similar way, the types of the target objects may influence the direction relation captured by the direction-relation matrix. While a target polygon must extend through at least one tile of the grid around the reference object, a line or a point may fall between the cracks. For example, a linear target object may be located exactly along the border between two neighboring tiles (Figure 3a) or a point may fall not only on the border between two neighboring tiles, but also coincide with the border of four tiles (Figure 3b). A reference grid that supports all cases for point, line, and polygon targets would require nine partitions with sixteen boundary components.

x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
TL	T	TR	R	BR	B	BL	L	DP
256	128	64	32	16	8	4	2	1

Figure 4: The nine-bit field for an element of the deep direction-relation matrix.

The neighbor code N is then the sum of bit numbers with non-empty slots (Equation 3). It ranges from 0 to 510. The value 0 indicates that the target object intersects neither with this direction partition nor with its boundaries. The value 1 captures a non-empty intersection between the target object and this direction partition. The only possible odd value is 1, as we record neighbor codes only if the object does not intersect with the direction partition.

$$N = \sum_{i=0}^8 2^i x_i \quad (3)$$

An important property is that each neighbor code can be uniquely decomposed into its contributing bits, which implies that the sum of the bit values is a loss-less encryption. For example, the neighbor code 170 can be obtained only for a tile with an empty direction partition (DP) and intersections with the left (L), bottom (B), right (R), and top (T) boundary parts (i.e., $2^1 + 2^3 + 2^5 + 2^7$).

To capture the cardinal direction, the boundary parts are analyzed locally for each of the nine direction partitions, yielding nine neighbor codes. When these neighbor codes are arranged in the same topological organization as the direction-relation matrix, this structure yields the *deep direction-relation matrix* (Equation 4).

$$D = \begin{bmatrix} N_{NW} & N_N & N_{NE} \\ N_W & N_0 & N_E \\ N_{SW} & N_S & N_{SE} \end{bmatrix} \quad (4)$$

4.1 Compatibility with Direction-Relation Matrix

Since the direction-relation matrix and the deep direction-relation matrix rely on the same 3x3 structure that refers to the same spatial subdivision, it is possible to compare the two models. Any differences between them are due to the neighbor codes.

The two models behave exactly the same if (1) the reference objects is a polygon or a line that is neither strictly horizontal nor vertical, *and* (2) the target object does extend up to the border of a direction tile without crossing it. For other configurations (Sections 3.1 and 3.2) the deep direction-relation matrix provides more detail if non-empty tiles border with the target object. For example, if a target object coincides with the MBR formed around the reference object (Figure 5a), then the eight direction tiles around the center have non-zero values in the deep direction-relation matrix (Figure 5b), while they are empty in the direction-relation matrix (Figure 5c).

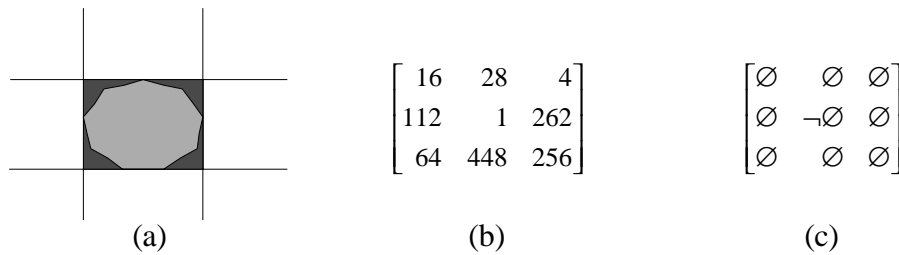


Figure 5: Example of a configuration in which the deep direction-relation matrix provides more detail than the direction-relation matrix: (a) the target object coincides with the MBR formed around the reference object with (b) its deep direction-relation matrix and (c) its direction-relation matrix.

These dependencies between entries in the two matrices applies in all situations, therefore, the entries of the direction-relation matrix can be derived from the deep direction-relation matrix. This mapping is straight forward as it benefits from the structure of the neighbor code: all even-numbered neighbor codes refer to empty intersections, while the only odd-numbered neighbor code (i.e., 1) stands for a non-empty intersection (Equation 5).

$$\phi = \begin{cases} \text{even}(N) \rightarrow \emptyset \\ \text{odd}(N) \rightarrow \neg\emptyset \end{cases} \quad (5)$$

4.2 Expressive Power for Reference Objects with Varying Dimensions

To assess whether the deep direction-relation matrix is a suitable direction model, it is necessary that the four types of reference frames (Section 3.2) are supported sufficiently, so that in each of them distinct direction relations are captured by different direction-relation matrices. We have chosen a set of representative examples. A more comprehensive and systematic evaluation is reported elsewhere [8].

Figure 6a-b shows two configurations, each with a polygon reference object and a target in the north-east tile. Their deep direction-relation matrices differ in one neighbor code, because the target

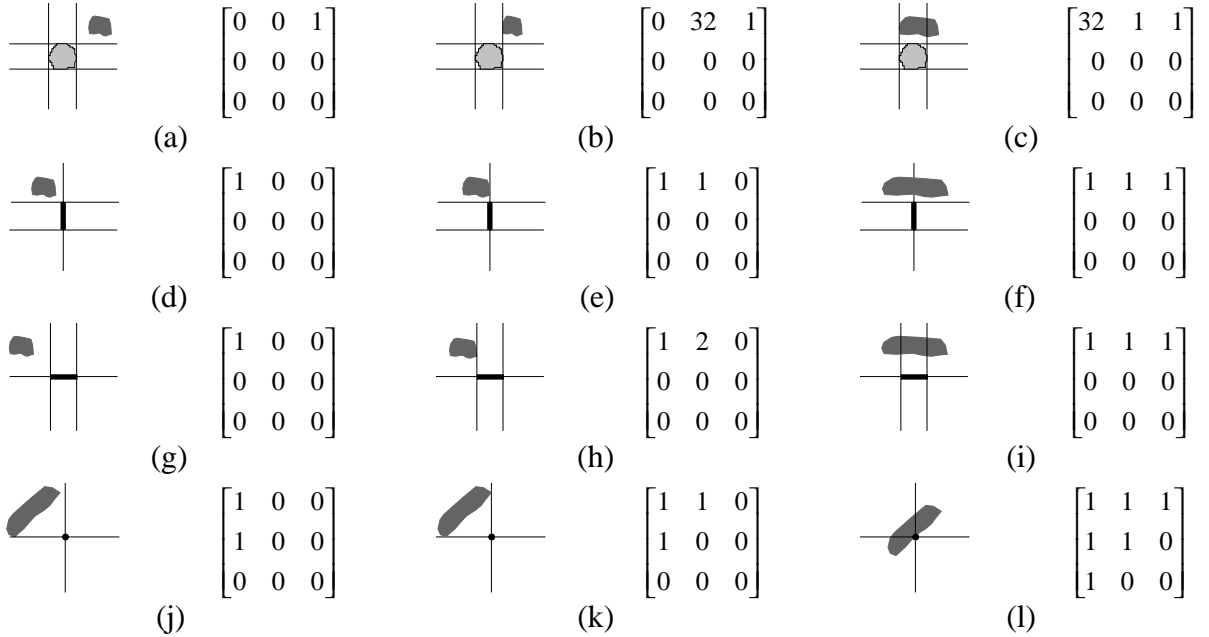


Figure 6: Configurations with (a-c) polygons, (d-f) vertical lines, (g-i) horizontal lines, and (j-l) points as reference objects.

in Figure 6b extends up to the most eastern part of the reference object. These two deep direction-relation matrices are significantly different, however, from the one in Figure 6c, where the target extends from the northeast tile through the northern tile up to the most western part of the reference object. The three configurations in Figure 6d-f have a vertical line as a reference object. The target region is located either completely in the northwest tile (Figure 6d), in the northwest tile up to the vertical reference line (Figure 6e), or extends from the northwest to the northeast tile (Figure 6e). Their deep direction-relation matrices capture these differences (without neighbor codes, because only three vertical partitions exist). Figure 6g-i shows the same locations for the target objects, but this time with respect to a horizontal reference line. Their three deep direction-relation matrices distinguish the direction relations correctly. The last three configurations in Figure 6j-l are with respect to a reference point. In Figure 6j the target extends through the northwest tile, up to the reference point; in Figure 6k the target has moved further east, up to the reference point; and in

Figure 6l the target has drifted towards the southeast such that it touches the reference point. In all cases, the deep direction-relation matrix captures these differences correctly.

4.3 Expressive Power for Target Objects with Varying Dimensions

A second test for the suitability of the deep direction-relation matrix is whether it distinguishes sufficiently direction relations for different types and locations of target objects. Figure 7a-f shows examples with point targets. While the configurations in Figure 7a, c, and f could be distinguished with the coarse direction-relation matrix alone, the configurations in Figure 7b, d, and e need the deep direction-relation matrix to describe the situations, because the target point coincides with the boundaries between tiles. The deep direction-relation matrix also makes a clear distinction between the three situations. In a similar way, the six direction relations in Figure 7g-n with a target line are distinguished uniquely by the deep direction-relation matrix. Only the relations in Figure 7h, k, and l would have been captured by the coarse direction-relation matrix, although certain details would have been overlooked, such as the target line running up to the northeast tile's corner in Figure 7h. The six configurations in Figure 7g, i, j, and m-o would have resulted in an empty direction-relation matrix, because all lines fall between the direction tiles. Figure 7m-n gives examples of deep direction-relation matrices that are filled with nine neighbor codes. Figure 7n is unique as it is the only configuration with the maximum matrix element in a deep direction-relation matrix. Finally, Figure 7o shows an areal target object that coincides with the MBR around the reference object.

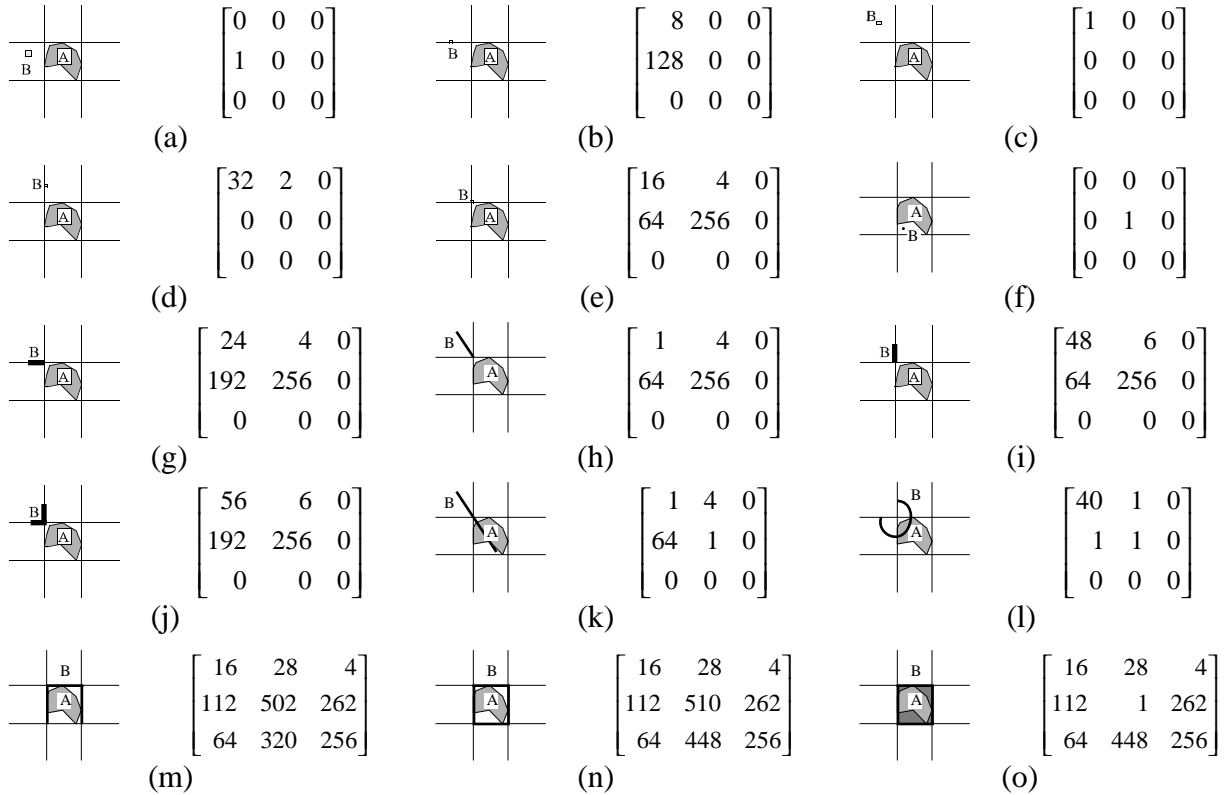


Figure 7: Configurations with (a-f) points, (g-n) lines, and (o) a polygon as target object.

5. Compatibility Across Different Levels of Detail

An important aspect of an integrated direction-relation model is that it captures compatible values for equivalent directions, irrespective of the objects' dimensions. Since such incompatibilities typically arise under generalization, we consider the direction relations between objects of different detail. We denote a direction relation at level of detail i with D_i , where i is an integer value.

Smaller values refer to more detailed representations, while larger values denote more generalized representations.

The direction captured by D_i is *compatible* with D_j , where $i > j$, if for each non-zero element in D_i the corresponding element in D_j is non-zero (Equation 6). It is a partial order relation, because it is reflexive (i.e., $compatible(D_i, D_i)$), transitive ($compatible(D_i, D_j) \wedge compatible(D_j, D_k) \Rightarrow compatible(D_i, D_k)$), and antisymmetric ($compatible(D_i, D_j) \wedge compatible(D_j, D_i) \Rightarrow equal(D_i, D_j)$).

$$compatible(D_i, D_j) \text{ if } \forall a, b : D_i[a, b] \neq 0 \Rightarrow D_j[a, b] \neq 0 \quad (6)$$

We demonstrate this property of the deep direction-relation matrix through a set of representative configurations. A comprehensive proof of the compatibility is available elsewhere [8]. Figure 8a shows a polygonal target, located in the north tile of the reference object. A generalization to a line results in a compatible direction relation if the line remains within the limits of the north tile (Figure 8b). Likewise, a generalization to a point results in a compatible direction relation as long as the point is located in the north tile. In all three cases, identical deep direction-relations would be obtained.

If the target region extends up to the reference object's most eastern point (Figure 8d), the generalization may generate a line (Figure 8e) or point (Figure 8f) that extends up to the northeast line. Their direction-relation matrices are compatible even if their entries and neighbor codes are not exactly the same. For instance, the direction to the generalized point (Figure 8f) is compatible with the direction to the target region.

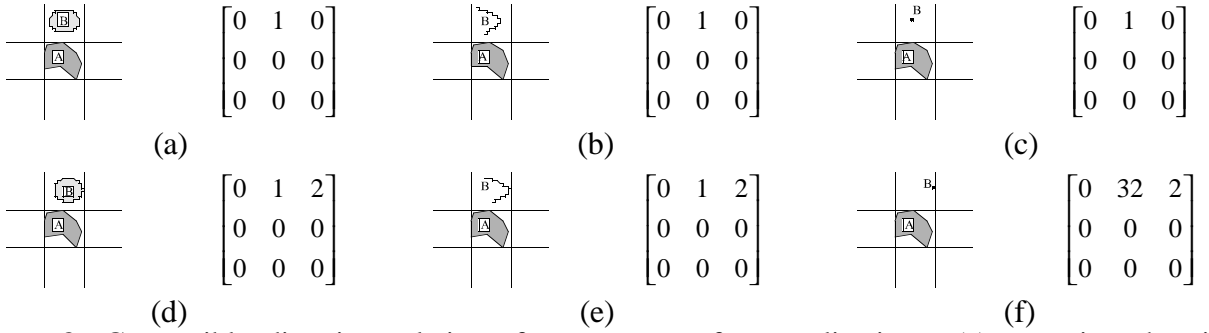


Figure 8: Compatible direction relations for two sets of generalizations: (a) a region that is generalized to (b) a line or (c) a point; and (d) another region that is generalized to (e) a line or (f) a point.

6. Conclusions

This paper presented an integrated model for cardinal direction that applies to arbitrary pairs of points, lines, and polygons. The model records directions based on the objects' geometries, not their approximations. It records equal matrices for the equivalent direction between pairs of objects, regardless of their dimensions. The results of spatial queries involving directions are compatible; therefore, the model is applicable across different levels of detail. The deep direction-relation matrix between a pair of objects is recorded once using the geometries of the objects, and all the spatial queries are processed based on the recorded matrices. This model can improve results of spatial queries involving directions in multi-resolution spatial databases. The integrated direction model frees the user from pondering about the dimension of the objects.

A comparison with the most common direction-relation model for extended objects, which is based on MBRs [14, 16], reveals the advantages of the deep direction-relation matrix. In the MBR approach, all objects that have their MBRs intersecting with the search rectangle are considered to be in the given direction; however, this technique may return wrong results when objects are not rectangular. Due to the MBR approximations, this method cannot consider the objects' shapes, which is an important factor in determining direction [1, 17]. The deep direction-relation matrix does not approximate shape and applies to all dimensions of objects. In the projection-based

direction partitions there are four types of objects: point, horizontal-line, vertical-line, and polygon. These four types of objects can be reference as well as target objects, which gives sixteen different types of pairs. Out of these sixteen types, the MBR-based model applies to only one of the sixteen types (i.e., a pair of polygons). For example, a horizontal-line target object can coincide with a grid line for a polygon reference object, but none of the 169 MBR relations can represent this relation. The deep direction-relation matrix, however, provides a unifying framework to represent direction for all sixteen pairs.

7. References

- [1] A. Abdelmoty, *Modelling and Reasoning in Spatial Databases: A Deductive Object-Oriented Approach*, Ph.D. Thesis, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, Scotland, 1995.
- [2] M. Bertolotto, *Geometric Modeling of Spatial Entities at Multiple Levels of Resolution*, Ph.D. Thesis, Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Genova, Italy, 1998.
- [3] B. Buttenfield, *Multiple Representations: Initiative 3 Specialist Meeting Report*, Technical Report, National Center for Geographic Information and Analysis, Santa Barbara, CA, 1989.
- [4] L. Cardelli and P. Wegner, On Understanding Type, Data Abstraction, and Polymorphism, *ACM Computing Surveys*, vol. 17, no. 4, pp. 471-552, 1985.
- [5] M. Egenhofer, Spatial SQL: A Query and Presentation Language, *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 1, pp. 86-95, 1994.
- [6] A. Frank, Qualitative Spatial Reasoning: Cardinal Directions as an Example, *International Journal of Geographical Information Systems*, vol. 10, no. 3, pp. 269-290, 1996.
- [7] J. Freeman, The Modelling of Spatial Relations, *Computer Graphics and Image Processing*, vol. 4, no. pp. 156-171, 1975.
- [8] R. Goyal, *Similarity Assessment for Cardinal Directions Between Extended Spatial Objects*, Ph.D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, 2000.
- [9] R. Goyal and M. Egenhofer, Cardinal Directions between Extended Spatial Objects, *IEEE Transactions on Knowledge and Data Engineering* (in press).
- [10] D. Hernández, *Qualitative Representation of Spatial Knowledge*, Lecture Notes in Artificial Intelligence, vol. 804, Springer-Verlag, New York, 1994.
- [11] R. McMaster and S. Shea, *Generalization in Digital Cartography*, American Association of Geographers, Washington, DC, 1992.
- [12] G. Miller, The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information, *Psychological Review*, vol. 63, no. pp. 81-97, 1956.
- [13] R. Newell, Practical Experiences of Using Object-Orientation to Implement a GIS, *GIS/LIS '92*, San Jose, CA, 1992, pp. 624-629.
- [14] D. Papadias and T. Sellis, Qualitative Representation of Spatial Knowledge in Two-Dimensional Space, *VLDB Journal*, vol. 3, no. 4, pp. 479-516, 1994.
- [15] D. Papadias and T. Sellis, A Pictorial Query-by-Example Language, *Journal of Visual Languages and Computing*, vol. 6, no. 1, pp. 53-72, 1995.
- [16] D. Papadias, Y. Theodoridis, and T. Sellis, The Retrieval of Direction Relations using R-Trees, *Database and Expert Systems Applications—5th International Conference, DEXA '94, Athens, Greece*, D. Karagiannis, ed., *Lecture Notes in Computer Science*, Springer-Verlag, New York, 1994, pp. 173-182.
- [17] D. Peuquet and C.-X. Zhan, An Algorithm to Determine the Directional Relationship Between Arbitrarily-Shaped Polygons in the Plane, *Pattern Recognition*, vol. 20, no. 1, pp. 65-74, 1987.
- [18] E. Puppo and G. Dettori, Towards a Formal Model for Multi-Resolution Spatial Maps, *Advances in Spatial Databases—Fourth International Symposium on Large Spatial Databases, SSD '95*, Portland, ME, M. Egenhofer and J. Herring, eds., *Lecture Notes in Computer Science*, Springer-Verlag, New York, 1995, pp. 152-169.
- [19] G. Retz-Schmidt, Various Views on Spatial Prepositions, *AI Magazine*, vol. 9, no. 2, pp. 95-105, 1988.

- [20] N. Roussopoulos, C. Faloutsos, and T. Sellis, An Efficient Pictorial Database System for PSQL, *IEEE Transactions on Software Engineering*, vol. 14, no. 5, pp. 630-638, 1988.
- [21] M. Schneider, *Spatial Data Types for Database Systems*, Lecture Notes in Computer Science, vol. 1288, Springer-Verlag, Berlin, 1997.
- [22] P. Svensson and H. Zhexue, Geo-SAL: A Query Language for Spatial Data Analysis, *Advances in Spatial Databases—Second Symposium, SSD '91*, O. Günther and H.-J. Schek, eds., *Lecture Notes in Computer Science*, Springer-Verlag, New York, NY, 1991, pp. 119-140.