Fuzzy Spatial Relationships and Mobile Agent Technology in Geospatial Information Systems

Frederick E. Petry [1 *], Maria A. Cobb [2], Dia Ali [2], Rafal Angryk [3], Marcin Paprzycki [2], Shahram Rahimi [3], Lixiong Wen [4], Huiqing Yang [3]

[1] Naval Research Laboratory
Mapping, Charting & Geodesy
Stennis Space Center, MS 39529
petry@eecs.tulane.edu

[2] Department of Computer Science & Statistics
University of Southern Mississippi
Hattiesburg, MS 39406-5106
firstname.lastname@usm.edu

[3] Center for Computational Science
University of Southern Mississippi
Hattiesburg, MS 39406-5106
firstname.lastname@usm.edu

[4] Department of Electrical Engineering & Computer Science
Tulane University
New Orleans, LA 70118-5674

Abstract. This chapter discusses an integrated work in the definition and implementation of sets of fuzzy spatial relationships concerning topology and direction. We present our basic approach to defining these relationships as an extension to previous work in temporal relations. We also discuss several extensions to this approach that include refinements and alternate definitions. Two implementations are also described, one in a C++, Oracle database environment and another utilizing the expert system shell Fuzzy Clips. Finally we discuss the integration of this querying approach in an agent-based framework. Agent technology has become a leading implementation paradigm for distributed and complex systems, and has recently garnered much interest from researchers in the area of spatial databases. Agents offer many advantages with respect to intelligence abilities and mobility that can provide solutions for issues related to uncertainty in spatial data, such as those of spatial relationships.

1 Introduction

The need to handle imprecise and uncertain information concerning spatial data has been widely recognized in recent years, e.g., [19], particularly in the field of geographical information systems (GIS). GIS is a rather general term for a number of approaches to the management of cartographic and spatial information. Most definitions of a GIS [16,22] describe it as an organized collection of software systems and geographic data able to represent, store and provide access for all

---

[*] On sabbatical from Tulane University, New Orleans, LA, USA.

forms of geographically referenced information. At the heart of a GIS is a spatial database. The spatial information describes the location and shape of geographic features in terms of points, lines and areas.

There has been a strong demand to provide approaches that deal with inaccuracy and uncertainty in GIS. The issue of spatial database accuracy has been viewed as critical to the successful implementation and long-term viability of GIS technology [19]. There are a variety of aspects of potential errors in GIS encompassed by the general term "accuracy." However, here we are only interested in those aspects that lend themselves to modeling by fuzzy set techniques.

Many operations are applied to spatial data under the assumption that features, attributes and their relationships have been specified a priori in a precise and exact manner. However, inexactness often exists in the positions of features and the assignment of attribute values and may be introduced at various stages of data compilation and database development. Models of uncertainty have been proposed for spatial information that incorporate ideas from natural language processing, the value of information concept, non-monotonic logic and fuzzy set, evidential and probability theory. For example, in [32] there are reviews of four models of uncertainty based on probability theory, Shafer's theory of evidence, fuzzy set theory and non-monotonic logic. Each model is shown as appropriate for a different type of inexactness in spatial data. Inexactness is classified as arising primarily from three sources. "Randomness" may occur when an observation can assume a range of values. "Vagueness" may result from imprecision in taxonomic definitions. "Incompleteness of evidence" may occur when sampling has been applied, there are missing values, or surrogate variables have been employed. An excellent collection of recent papers on vague boundaries in spatial applications can be found in [4]. Various topics in the volume include some areas of particular interest such as topological relations and indeterminate boundaries, data models for indeterminate objects and fields, and vague shape models.

Robinson [29,28,27] has done early extensive research on fuzzy data models for geographic information. He has considered several models as appropriate for this situation—the two early fuzzy database approaches using simple membership values in relations [18,2], and a similarity-based approach [3]. In modeling a situation in which both the data and relationships are imprecise, he assesses that this situation entails imprecision intrinsic to natural language which is possibilistic in nature [37].

There have been several recent special sessions on fuzzy sets and GIS at FUZZ-IEEE’98, IPMU’98 and NAFIPS’99. A collection of such work has been collated into a special issue of F*uzzy Sets and Systems* [12]. In the issue are found a number of approaches to the use of fuzzy sets and spatial data. Relevant selections include fuzzy objects for GIS [13], landform classification with fuzzy k-means [5], and fuzzy spatial queries [33].

In this chapter we describe the development of an approach for fuzzy querying using spatial relations, focusing especially on directional querying. We first provide an overview of the background work in this area, including, in particular, spatial relationships using Minimum Bounding Rectangles (MBRs) and fuzzy set-based spatial data approaches.

Next we describe our basic approach [11] using a variant of the spatial intervals and an underlying model called an abstract spatial graph (ASG) to support fuzzy querying. Following, we consider further development of the approach, such as the use of finer partitions know as MRRs to enhance querying. Then we discuss some modifications to resolve certain outlying cases that produce anomalous query results. Two implementations are also described, one in a C++, Oracle database environment and another utilizing the expert system shell, Fuzzy Clips. Finally we discuss the integration of this querying approach in an agent-based framework.

2 Background

Relevant background research includes various aspects of spatial reasoning, work on directional and topological relationship representation and the incorporation of uncertainty and fuzzy querying.

A basis for many researchers' approaches is the extension of Allen' s temporal relations [1] to two or more dimensions for spatial reasoning. Examples of how this has been done can be found in [20,31,24,23] to name a few. For each of these, the approach taken is somewhat different, based on the intent of the work. However, the concept of representing a 2-D object as a set of two intervals, an x and a y, and of having the resulting spatial relationship consist of some combination of the component 1-D relations seems to underlie most.

Also relevant is [14], which presents a unified framework for approximate spatial and temporal reasoning using topological constraints as the representation schema and fuzzy logic for representing imprecision and uncertainty. The application of the resulting fuzzy representation to each of Allen' s interval relationships is developed as the possibility of the occurrence of the conditions of the original definition. A different approach based on statistical methods for representing and deriving topological relations is given in [35]. The relations used are those in [15] restricted to 1-D, for which Winter introduces a partial ordering based on Galtons' ordering of topologic relations for space and time sequences [17]. The derivation of uncertain topologic relations is treated as a classification problem. As such, specific conditions for deriving relationship probabilities based on the testing of dominant vs. dominated relations are presented.

Additionally, all of this work appears to utilize the fact that Allen's interval relations and corresponding logic have relevance in two dimensions to two extremely significant areas in spatial reasoning: topology and direction (orientation). Thus, the various extensions to Allen's work have provided sound foundations from which to launch work in qualitative spatial reasoning. Sharma's work, in particular, takes advantage of the dual benefit of the model by showing how inferences can be made over the composition of topological and directional information [31]. This extension includes a mapping of the temporal relations onto the eight mutually exclusive binary topological relations of the 4-intersection [15], a generic model that defines topological relationships through the intersections of boundaries and interiors of point sets.

The approach we have taken, as well as that of Nabil [24], Sharma [31] and Clementini [8], relies upon the use of MBRs as approximations of the geometry of spatial objects. An MBR is defined as the smallest X-Y parallel rectangle which completely encloses an object. This is a typical MBR around an irregular object such as the lake shown in the figure. The use of MBRs in geographic databases is widely practiced as an efficient way of locating and accessing objects in space. In addition, numerous spatial data structures and indexing techniques have been developed that exploit the computationally efficient representation of spatial objects through the use of MBRs [21,30]. Recently, Papadias and Theodoridis [25] have considered various indexing issues (R-trees, KDB trees etc.) to process topological and directional queries using MBRs. Their particular focus has been experimentation with alternative indexing to provide query optimization.

3 Fuzzy Directional Relationships and Querying

The core of the approach we shall describe is directly dependent upon the definitions of binary relationships between two-dimensional objects. For our purposes, we are assuming that images have been segmented and labeled, and that objects representing features have had MBRs assigned. We utilize an extension into the spatial domain of Allen's temporal relationships [1] where he defined a set of thirteen relationships that completely represents any relationship that can exist between two one-dimensional (temporal) intervals. These relationships are *before, equal, meets, overlaps, during, starts* and *finishes,* along with their inverses.

The spatial extension is as follows: given the MBRs of two objects, the binary relationship between the objects in both the horizontal and vertical directions can be completely defined by a tuple, $[r_x, r_y]$, where r_x is the one of Allen's temporal relations that defines the interaction of the object MBRs in the x direction, and r_y represents the same for the y direction. This results in a total of 85 possible relationships—49 base relations and 36 inverse relations.

The formal definitions for each of the relationships is given in terms of a set of constraints based on corner positions, each of which must hold between the MBRs for each object. For example, for the case of A [finishes,starts] B , the definition is given as: $\{ B_{xl} < A_{xl} < B_{X2}, A_{X2} = B_{X2}, B_{yl} < A_{y2} < B_{y2}, A_{yl} = B_{y1} \}$, where (xl,y1} and {x2, y2} represent the lower left and upper right corners, respectively, of the MBRs. In Figure 1 is an example set of object MBRs and a subset of the existing relationships between. This set of MBRs and relationships will be used to demonstrate the modeling and query framework.

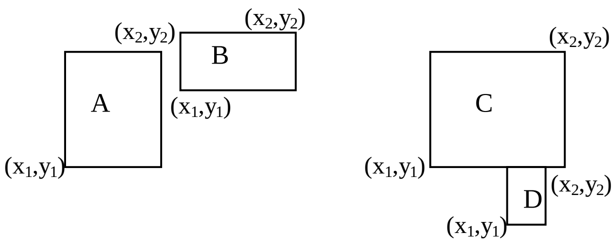


Fig. 1. MBR relationships.
A [before, overlaps] B; B [before, overlaps $^{-1}$] C; D [during, meets] C.

These spatial relationships can be used for qualitative topological relationship definitions. We include in our set of "qualitative topological relationships" definitions that are intuitive and useful from a user's perspective, but which do not necessarily satisfy the property of topological invariance. Such definitions evolve by recognizing that logical groupings of the relationships can be used to define common topological concepts. The partitioning of the relationships in this manner can result in as many or as few groupings as desired. Such partitionings can be represented as sets, each of which is then associated with a linguistic term.

The implication of this is that any relationship that is contained within a particular set can be considered an independent representation of the corresponding qualitative topological relationship, represented by its linguistic term. Some examples are shown below using the notation of representing one of Allen's relations by its initial letter.

surrounded-by := { [dd] }

partially-surrounded-by := { [df] [fd] [do] [ds] [od] [sd] [do^{-1}] }

overlapped-by := { [oo^{-1}] [os^{-1}] [of^{-1}] }

Such definitions serve merely as examples, and may be redefined according to individual needs.

Next, the basic relationship definitions can be used in a similar manner for defining directional relationships. Specifically, the relationships with which we concern ourselves here are the cardinal directions and refinements: N, S, E, W, NE, SE, SW, NW. Given that the context for consideration is that of directional relationships between two-dimensional objects, it is apparent that a simple, crisp representation of direction is not adequate for supporting any but the simplest of query capabilities. Given the spatial extent of two-dimensional objects, it is very likely that in any one case, more than one of the eight directions listed above will apply, to either a greater or lesser degree. Therefore, we have developed a method for defining directional relationships that would allow for fuzzy querying of *any* of the directional relationships that exists between two objects.

The concept of *object sub-groups* is used as a basis for determining the set of directions that defines the directional relationship between two objects. Object sub-groups, similar in nature to ortho-relational objects described in [7], are partitions of the MBRs created by the extension of the edges of one or both of the MBRs involved in one of the 85 previously defined relationships. In those cases for which no extensions intersect the partner MBR, the entire object is considered to be an object sub-group. Additionally, any overlapping portion of the MBRs is defined as an object sub-group. Examples of MBR partitioning into object sub-groups for some example relationships are shown in Figure 2.

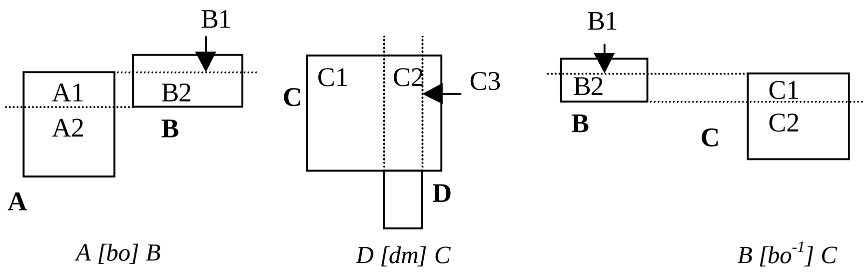


Fig. 2. Examples of object sub-group partitionings.

A *direction set is* then derived from this partitioning by identifying all possible directions that can be associated with any of the object sub-groups of A with respect to all of the object subgroups of object B. Given this, the direction sets corresponding to the object sub-groups shown in Figure 2 are:

{W, SW} for A [bo] B; {W, NW} for B [bo^{-1}] C; {S, SW, SE} for D [dm] C.

Direction sets were derived in this manner for each of the 85 relationships. The result was the creation of neighborhoods of sizes two and four for which the direction sets were equivalent for the constituent relationships. Direction sets for the relationships grouped by neighborhoods, which are considered as *equivalence* *classes* for directional relationships. All relationships within an equivalence class are represented by the corresponding direction set, and are considered equivalent with respect to direction. The creation of equivalence classes within and between the base and inverse relationship groups results in 27 possible direction sets.

Definitions for directions can now be defined in a manner analogous to the way in which qualitative topological relationships were defined earlier. The definition for any particular direction includes the set of all relationships containing that direction as a member of its direction set. The definition for the direction East is shown below as an example.

$$E ::= \{[dd], [df], [fd], [do], [ds], [ff], [d=], [fo], [fs], [f=], [dd^{-1}], [do^{-1}], [ds^{-1}], [fd^{-1}], [df^{-1}], [fo^{-1}], [fs^{-1}]\}$$

The basic relationship definitions and their use in defining relevant directional and qualitative topological relationships provide a framework for the *abstract spatial graph* (ASG), a spatial data structure specifically designed to retain orientation and topological information with respect to two-dimensional objects, and to provide information to support fuzzy querying capabilities on these relationships

The first step in constructing the ASGs is to categorize the original 85 relationships according to the level of interaction of the MBRs into four distinct categories: disjoint, tangent, overlapping and containment. Similar generalizations of relationships have been proposed, for example, in [23]. These categories provide sufficient distinction for formulating the definitions for the concepts of *reference areas* and *reference points* necessary for establishing an ASG for a particular binary relationship.

The general concept of a reference area is that of some region which is either common to two objects, or which can be intuitively derived from their relationship. A reference point is taken to be the centroid of the reference area. It is at the reference point that a directional axis is centered for the purpose of constructing the ASG. The four categories of relationships, along with definitions and examples for reference areas and reference points for each are shown in Figure 3. In the figure we see that for the disjoint relationship the reference area is the region between the two objects bounded on two sides by the neighboring sides of the MBRs and on the other two sides by appropriate extensions of the MBR sides (horizontally, from the leftmost object). The centroid of the reference area is the reference point in this case. The reference area, or line in the case of tangency, is the common tangent line segment and its center is the reference point. For overlapping figures, the reference area is the area of overlap, and for the case of the containment relationship, the reference area is the MBR of the contained object. In both of these cases the reference point is the centroid of the reference area.

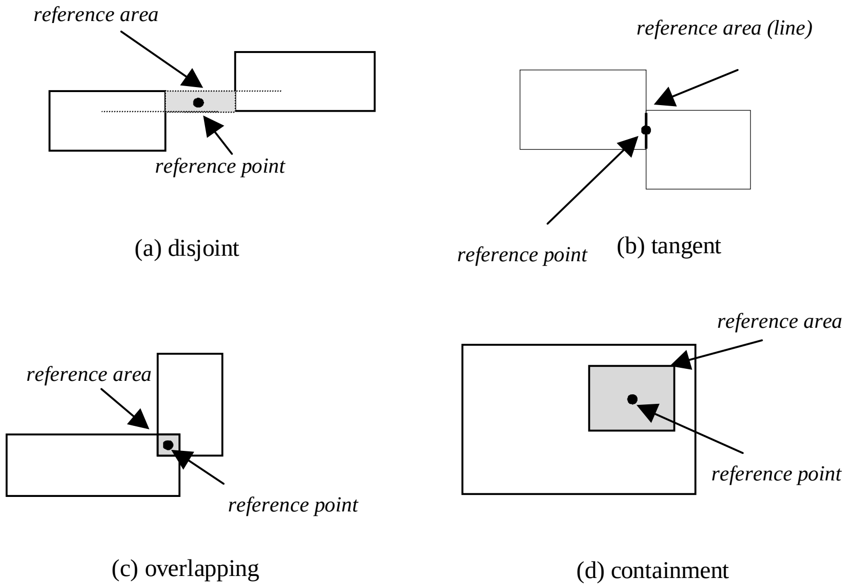


Fig. 3. Reference areas and reference points.

Again, the construction begins by centering a directional axis containing line segments for each of the eight compass points at the reference point of the relationship. Then, for each object sub-group through which an axis passes, a node for that object is placed on the graph at an orientation corresponding to that of the axis that crosses the object sub-group.

To avoid "cluttered" graphs that contain many nodes with low information content, it is desirable to set a minimum threshold on axis lengths (defined as the length of the axis segment from its entrance into an object sub-group to its exit from the same sub-group) such that for any situation in which the axis length is below the set threshold, no node is constructed. Such a threshold is obviously application dependent, and possibly even variable within an application.

Pictorially, nodes are each placed an equal distance from the origin. Different node representations (e.g., color-coded) are used for nodes representing object sub-groups belonging to different objects. An arc is then drawn to connect all of the nodes that represent the same object. The origin symbol differs depending upon the relationship's membership in one of the four basic categories.

In addition to providing information directly relevant to the representation of the abstract spatial graph, we also needed to represent ancillary information that can be used for fuzzy query inferences. This information is represented in the form of node "weights" that can be used for defining fuzzy topological and directional qualifiers for use with a fuzzy query framework.

Calculation of weights uses both the areas of object sub-groups and the lengths of axes that pass through object sub-groups. Three different types of weights are computed: *axis weights, area weights* and *node weights.* Each of these is derived in a specific manner designed to support a given objective for fuzzy querying.

Axis weights are an intermediate step for calculation of node weights. First, all axes whose lengths are less than some set minimum threshold are discarded, ensuring that we are dealing only with those axes whose lengths are significant. Then, the longest of the remaining axes is normalized to 1, and the weight of each of the other axes is computed as a ratio of its axis length to the length of the longest axis.

Area weights are also used in the calculation of node weights; however, these also have significance of their own. Area weights are calculated for each object sub-group and are defined as the ratio of area occupied by the object sub-group to the area of the entire object, as defined by the MBR. Finally, a total node weight is derived by multiplying the axis weight by the area weight for each node of the ASG.

Area weights and total node weights are stored for each node of the ASG, with the only exception being the origin node, which stores only an area weight. Since the origin node represents the reference area itself, axis length is not a reasonable consideration for the reference area object sub-group. Therefore, a node weight for the origin node can not be computed, and we allow the area weight to serve as the total node weight also.

The area weights and total node weights of ASGs directly support fuzzy queries regarding qualitative topological and directional information in two specific ways. Area weights provide an indication of the degree to which an object participates in a qualitative topological relationship. By mapping ranges of area weights to linguistic qualifiers such as *some, most,* etc., fuzzy information such as "some of object A overlaps most of object B," can be determined.

Total node weights, on the other hand, are used to indicate the extent to which one object can be considered to lie in a certain direction with respect to a second object. Again, ranges of weights can be correlated to linguistic terms, e.g., *slightly, mostly,* to provide qualifiers for directional orientation. Then, for example, one could determine that, while object A is *slightly southwest* of object B, it is at the same time *mostly west* of object B.

The preservation of all directional information regarding two objects, along with the use of total node weights, allows users to obtain a complete conceptualization of directional relationships between the objects. Furthermore, the calculation of the total node weight as the product of the axis weight and area weight ensures against bias in those cases, for example, where the object sub-group associated with a directional axis is very large (increasing the weight for that direction), but for which the axis weight is very small (indicating a weaker association for that sub-group/direction pair than for others for the same object).

By using the following such assignment for area weights:

{all(96 - 100 %), most(60 - 95 %), some(30 - 59 %), little(6 - 29 %), none(0 - 5 %)}

and node weights:

{directly(96-100 %), mostly(60-95 %), somewhat(30-59 %), slightly(6-29 %), not(0-5 %)}

we can determine for our example of Figure 1 that:

1. B is *mostly* west of C
2. *Little* of B is northeast of A
3. D is *directly* south of C
4. C is *slightly* southeast of B

4 Extensions to the Model

4.1 Extensions to the Standard MBR Representation

The ASG model utilizes MBRs to represent features as described above, while algorithms for computing relationships for the model are designed in such a way as to minimize the potentially negative impact of the use of MBRs. Additionally, the ASG model extends the use of rectangular boundaries as representations of sub-objects within the MBRs.

In keeping with generally accepted practice, both MBRs and the sub-rectangles utilized in the ASG model retain x-y-parallelism; however, in this section we explore several variations on this traditional approach and investigate the respective implications to the modeling of relationships. Our goal in investigating alternative representations for geometric properties of spatial features is threefold: (1) alleviate or significantly decrease anomalies in topological relationship determination based on MBRs; (2) improve accuracy in determination of directional and topological relationships between representations; and (3) maintain, as much as possible, computational efficiency in processing concerning relationship categorization and querying.

First, we consider the implications of partitioning MBRs into sets of rectangles, essentially resulting in a gridded surface which is an approximation we call *Multiple Rectangle Representation*, or MRR. Three variations on MRRs which we analyze in this section include (1) a uniform MRR, (2) a non-uniform, congruent MRR, and (3) a non-uniform, non-congruent (irregular) MRR. All three variations of MRRs result in a finer approximation of the object's true geometry than do MBRs, while maintaining a basic regular, rectangular structure for which computationally efficient methodologies have been developed.

The first variation can be viewed as the imposition of a grid of any level of resolution upon an object, after which any of the rectangles not actually intersecting with a part of the object is discarded. Two cases include one in which grids of the same resolution are used for both objects participating in a relationship, as well as the case in which grids of different resolutions are used. Figure 4 shows a simple example of the use of uniform MRRs. The dotted line shows the original boundaries.

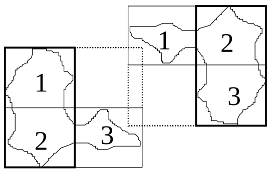


Fig. 4. Uniform MRRs used for object representations.

Now we consider enhancements to the ASG to accommodate the use of MRRs for relationship determination. We begin with the supposition that the set of ASGs is a closed set, such that any modification to the way in which relationships are defined does not result in any new ASG. It is apparent, however, that the way in which the ASGs themselves are defined for MRRs must be modified to take advantage of the more accurate representation.

We do this by first computing a set of ASGs—one ASG for every combination of relationships between sub-rectangles of both objects' MRRs. This results in a set of ASGs for each binary relationship, S ={A_{ij} | 1 <= i <= n, 1 <= j <= m, where n = number of subrectangles in 1st object, m = number of subrectangles in 2nd object}. For example, in Figure 6 are two objects with simple, uniform MRRs. The resulting set of relationship ASGs for this example is:

S = {A_{11} = [bo], A_{12} = [bo], A_{13} = [bo^{-1}], A_{21} = [bb],
A_{22} = [bb], A_{23} = [bo], A_{31} = [ob], A_{32} = [bb], A_{33} = [bo]}.

An approach for utilizing this information set to gain a more accurate picture of the relationship under consideration is to first associate a count with every distinct relationship appearing in the set. For our example, this would yield S′ = {([bo], 4), ([bb], 3), [bo^{-1}], 1), ([ob], 1)}. These counts divided by the total number of sub-rectangle relationship combinations (9 in the example) are considered as membership values in fuzzy relationships. Rather than having a single, crisp binary relationship as was the case in the original ASG model, we now have a *set* of ASG relationships, each member of which a given binary relationship belongs to some degree. For example, the fact that [bo] and [bb] appear as the predominant relationships for Figure 6, along with the fact that [bo^{-1}] and [ob] exist, although to a much lesser degree, conveys a substantially more significant amount of information than does the original [oo] designation, which in this case is also inaccurate with respect to the contained objects' relationship due to the topological inconsistency problem associated with MBRs.

We then take this a step further by associating these membership values with the higher-level relationships defined in the previous section [10]. Because these relationships represent a mutually exclusive, total partitioning of the basic relationships, a mapping from the members of S to these relationships will result in at most the same number of relationships as the number of members of S. However, it is more often the case that fewer high-level relationships result. This is because (1) such relationship definition sets are often composed of basic relationship neighbors, and (2) the use of MRRs in the manner described necessarily implies relationship categorizations for neighboring sub-rectangles, resulting in neighboring relationships. In our example, each of [bo], [bb], [bo^{-1}], and [ob] appears in the *disjoint* relationship, leading to the invariable conclusion that the two representations are indeed disjoint. This approach eliminates the need to compute the set of weights for ASG nodes as was done in the original model, as similar information is now maintained in S′.

The application of non-uniform, congruent MRRs can be understood as an analog to a quadtree decomposition commonly performed for spatial indexing purposes. The approach begins with standard x-y-parallel MBRs, upon which a quadtree-like decomposition is performed, with the MBR being divided into four equivalent rectangles, any or all of which can then be divided similarly, continuing until as fine a partitioning as desired is achieved. At that point, any rectangles not actually containing a part of the object are discarded. We say the rectangle sets are *regularly hierarchical* because the level of detail (size) of the rectangles can vary across different parts of the object so as to achieve a desired level of representation, while each larger rectangle is exactly a multiple to the fourth of any of the smaller rectangles of the object. An example of this type of MRR is shown by the grayed area in Figure 5.

This type of boundary approximation more accurately represents the objects' geometry in comparison to either MBR or uniform MRR representations. While maintaining the greatly reduced incidence of topological inconsistencies between true and approximate boundaries illustrated in the uniform MRRs, additional levels of detail have been added that allow for more accurate relationship determination. Since the areas are still rectangular decompositions, computational issues remain simplified compared to boundary representations.

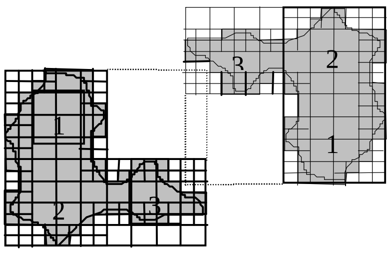
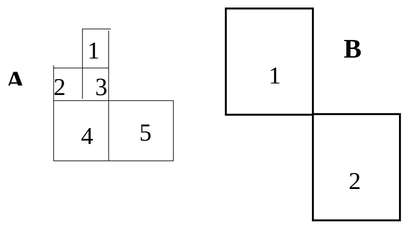


Fig. 5. A non-uniform congruent MRR.

The most significant issue that must be addressed concerning fuzzy relationship modeling for this approach is the manner in which the differently sized rectangles are assessed as contributors to overall relationship determination. This can be handled by extending the approach used for uniform MRRs. While for uniform MRRs it was sufficient to use the mere existence of the basic relationships between sub-rectangles as equal factors in fuzzy relationship categorizations, we must now compute a weight analogous to the ASG node weights for each sub-rectangle relationship to achieve a level of normalization for "combining" the individual relationships into one.

Recall that the area weights are computed as the ratio of the MBR sub-object areas to the total MBR area, and that these weights are used to identify the extent to which an object participates in a given relationship. Using this same approach for non-uniform congruent MRRs—calculating a weight as the ratio of a sub-rectangle's area to the combined area of all sub-rectangles containing a part of the object—we achieve a level of normalization for use of differently sized rectangles.

For each applicable relationship, the weights for every different sub-rectangle of each object for that relationship are summed, resulting in a value ≤ 1. Whenever one sub-rectangle participates in the same relationship with multiple sub-rectangles of the second object, that sub-rectangle's weight is only counted once. The resulting sums for a relationship for the two objects are then multiplied, yielding a weight for that relationship. For example, consider the case illustrated in Figure 6.



$A1_W = .1$; $A2_W = .1$; $A3_W = .1$; $A4_W = .35$; $A5_W = .35$; $B1_W = .5$; $B2_W = .5$

Fig. 6. Non-congruent MRRs with area weights.

For this example, we have the following set of relationships:

A \ B	1	2
1	[bd]	[bb^{-1}]
2	[bd]	[bb^{-1}]
3	[bd]	[bb^{-1}]
4	[bo]	[bo^{-1}]
5	[bo]	[bo^{-1}]

By summing and multiplying the area weights in the manner described earlier, we obtain the following :

$$[bd]_W = .15; [bo]_W = .35; [bb^{-1}]_W = .15; [bo^{-1}]_W = .35.$$

This shows that the relationships [bo] and [bo^{-1}] are weighted more heavily, primarily due to the larger areas of sub-rectangles 4 and 5 of object A. These weights can then be associated with the higher level relationships in the same manner as was shown for the uniform MRRs.

4.2 Geometric Modeling Capabilities

The use of MRRs for the ASG model has several implications. First is that the partitioning of MBRs in the ways described has *no effect* on the basic relationship definitions. Any relationship that originally held between two MBRs remains valid for the resulting MRRs, because minima and maxima for the objects do not change; therefore, any partitioning of the MBRs, regardless of granularity, will not affect the basic relationship between the geometric representations of the objects.

To support this statement, we begin by observing that the abstract spatial graphs can be arranged in a matrix according to the concept of *conceptual neighborhoods* with respect to the relationships that are represented. That is, each of the relationships that borders a particular relationship in both the horizontal and vertical directions can be derived from the given relationship without transitioning through any other relationship state. If a transition to an immediate neighboring relationship is impossible, then it follows that a transition to any other relationship is also impossible. We illustrate this by first examining the example shown in Figure 7, which represents a portion of the complete relationship matrix.

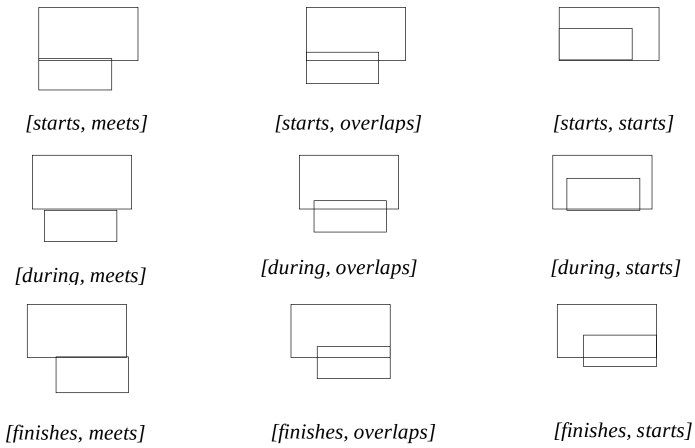


Fig. 7. [during, overlaps] relationship and conceptual neighbors.

Now we show that the enhanced accuracy of the boundaries for the MRR approximation model provides additional information in the way of geometric refinements for the relationship definitions. While there originally was only one geometric model for each relationship, there is now an infinite *set* of such models that correlates to any given relationship. Simple examples for a standard MRR approach are shown in Figure 8. Internal rectangle boundaries are omitted for clarity.

From this illustration, one can see that any selected MRR partitioning for a pair of MBRs will never cause a transition to one of the neighboring relationships. Therefore, we are able to operate under the assumption of a closed set of relationships for which the ASG model and query framework hold.

Figure 8 also demonstrates the advantages associated with the use of MRRs over MBRs: (1) the ability to better represent correct topology, and (2) the ability to make finer distinctions between geometric relationships. The first of these is illustrated in Figure 8(a) which shows that the two objects could not possibly overlap, while the original MBR representation shows an overlap. The second advantage can be seen in Figures 8(a)(b) and 8(c)(d) which show the same topological relationships—disjoint and overlaps—but for which geometric distinctions exist which provide additional information for spatial analysis.

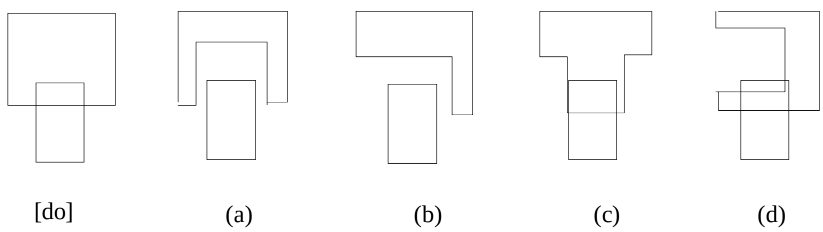


Fig. 8. A set of geometric representations for the [do] relationship.

4.3 An Extension for Expert System Implementation

Here we describe an implementation based on the C Language Integrated Production System (CLIPS). CLIPS is an expert system tool that provides a complete environment for the construction of rule and/or object-based expert systems [9,36]. Because of its portability, extensibility, capabilities, and low-cost, CLIPS has received widespread acceptance throughout the government, industry and academia.

Three rule sets are used to represent the basic structure of this model. Recalling that each object is represented by an MBR, the smallest definition of which is two diagonal points of the box, we consider two objects:

$$A\ (A_{x1}, A_{y1})\ \ (A_{x2}, A_{y2})$$

$$B\ (B_{x1}, B_{y1})\ \ (B_{x2}, B_{y2})$$

Now, taking one point from each object, that is,

$$(A1, A2) = (A_{x1}, A_{x2}) \quad \text{or} \quad (A_{y1}, A_{y2})$$

$$(B1, B2) = (B_{x1}, B_{x2}) \quad \text{or} \quad (B_{y1}, B_{y2}),$$

we will present the implementation process, beginning with definitions of three essential rule sets.

Rule Set 1: Define a set of non-ambiguous relationships.

Considering one direction, the spatial relation between object A and object B can be defined as in Figure 9. This is a partial set of the rules needed in actual CLIPS format.

```
1. IF < A2<B1 >    THEN    < before >
   IF < B2<A1 >    THEN    < before^-1 >
3. IF < A1<B1<A2<B2 > THEN < overlap >
   IF < B1<A1<B2<A2 > THEN < overlap^-1 >
```

Fig. 9. CLIPS rules for four non-ambiguous relationships.

Simply, this rule set can be expressed as:

$$r_x = (b, m, o, f, d, s, =, b', m', o', f', s').$$

In the y-direction, the same rules can be applied. Moreover, there are two additional rules that apply:

1. $A(r_x^{-1}, r_y)B = B(r_x, r_y^{-1})A$
2. $A(r_x^{-1}, r_y^{-1})B = B(r_x, r_y)A$

Rule Set 2: Define a set of topological relationships.

Based on the eighty-five basic relationships, the full topological relation set is:

T ={disjoint, tangent, surrounded-by, partially-surrounded, surrounded-by, partially-surrounds, overlapped-by, overlaps, x-subspace, y-subspace, y-subspaced-by}

Figure 10 shows a subset of the rules for topological relationships.

```
IF <dd> THEN  <A surrounded-by  B>

IF <oo'|os'|of'> THEN  <A overlapped-by B>
```

Fig. 10. Topological relationship rule set example.

Rule Set 3: Define the set of directional relationships.

Figure 11 shows one of the rules for directions in CLIPS.

```
IF   <dd|df|fd|do|ds|ff|d=|fo|fs|
      f=|dd'|do'|ds'|fd'|df'|fo'|fs'>

THEN  < A East B >
```

Fig. 11. Example CLIPS rule for direction.

Topological relations have been found to be useful for increasing the speed of spatial queries [34]. Therefore, let us analyze the geometric characteristics of topological relationships. Except for the disjoint relation, all other relations have a similar geometry; that is, the reference area is part of both objects involved. Thus, the original topological relation set can be reduced or reclassified to a binary topological set:

$$T \rightarrow T' = \{\text{disjoint, connected}\}$$

This new topological relation set will be used in the CLIPS implementation.

For convenience of implementation and further investigation, the ASG is modified by mapping topological relationships to 9 nodes for both objects. Similarly, each node has associated weights. However, the weight in a node may be null depending on the different topological relations. Since each object is associated with its 9 nodes, it is not necessary to keep information as to whether a node belongs to object A or object B in the implementation. This provides a flexible structure for fuzzy querying.

4.4 A CLIPS Implementation

Now we illustrate how CLIPS can be used to implement the binary spatial relationships given earlier. Because of the amount of computation involved in implementation, we can take advantage of the *deffunction* construct that allows the addition of new functions without having to recompile and relink CLIPS. Several user-defined functions are written by using the CLIPS *deffunction* construct, which can be executed by CLIPS interpretively.

As a rule-based shell, CLIPS stores knowledge in rules, which are logic-based structures. In the implementation, the basic three rules are defined by using *defrule* constructs. They provide the basic spatial information such as, *Object A is disjoint from Object B,* or *Object A is West of Object B.* For fuzzy querying purposes, extra functions and rules are defined that will support fuzzy querying. The implementation is directly dependent upon the reduced topological relation set and modified ASG mentioned above.

The facts are the critical resources for the querying. All details for binary spatial relations are contained in *deftemplate* facts. The type of information stored in the database includes the positions of two objects, the reference object, non-ambiguous relations, and topological and directional relationships. The corresponding data structures are declared by using *deftemplate* syntax.

To represent 2-D relations extended from Allen's relations, the *deffunction* construct in CLIPS is utilized. With this construct, a new function that implements Allen's relations in 1-D is defined directly in CLIPS. The rules that define a set of non-ambiguous relationships are built by a *defrule* construct.

The basic queries are based on the primary topological set and directional set. In this kind of querying, the degree to which one object lies in a particular direction with respect to a second object is not of concern. Figure 12 shows part of the CLIPS rule structures for directional relationships.

4.5 Fuzzy Querying of Binary Spatial Relationships

Based on the new topological relation set and modified ASG data structure, we defined three rules and four functions to support the processing of fuzzy queries. Query processing strategies are described as follows:

<u>Step1.</u> Find the reference area.
Fuzzy variable *weights* store all fuzzy query information. In order to get weights for each node in the ASG, a reference area must first be found. The reference area is also treated as an MBR object. We take two middle points among the four points in each direction as the reference object position. It can be represented as $R = (Rx_1, Ry_1)\ (Rx_2, Ry_2)$.

```
(defrule define-directional-relation
         (relationship (object1 ?A&A)
         (relations ?r) (object2 ?B&~A))
=>
 (if (numberp (member$ ?r (create$ od of
                sd sf dd df fd  ff =d =f
                ob' om' oo' os' …… )))
  then  (bind ?dr1 North))

               . . . . . .
 (loop-for-count (?count 1 8) do
    (bind ?dr (nth$ ?count (create$ ?dr1
              ?dr2 ?dr3  …… ?dr7 ?dr8)))
    (if (numberp (member$ ?dr (create$
                  North East …… West )))
     then
        (assert (directional-relationship
            (object1 ?A)(d-relation ?dr)
              (object2 ?B)) ) ) )    )
```

Fig. 12. Defrule for directional relationship.

Step2. Calculate weights.
Based on the binary topological relations, a general method was developed for connected relations. For example, **NW_area = (Rx_1 – Ox_1)(Oy_2 – Ry_2)**, where R represents the reference object, and O represents the one of two objects investigated. By adding some constraints, the general method for connected relations can also be applied to disjoint relations.

Given two objects and their reference object, the *weights* function maps the object sub-group into 9 nodes for each object, and calculates the area weights and node weights. The CLIPS program passes nine arguments to *weights* function, that is, one for object identifier, four for object position, and four for reference position. The function asserts area weights to the corresponding nodes for fuzzy querying.

Step3. Get qualifier to implement Fuzzy querying.
To provide support for fuzzy query processing, the fuzzy variable *weight*s is assigned to the corresponding linguistic terms qualifier. The *fuzzyTq* function defines the topological qualifiers that represent the linguistic terms for area weight. Similarly the *fuzzyDq* function defines the directional qualifiers that represent the linguistic terms for node weight.

The fuzzy set for topological qualifiers is described below:

{all (0.96 – 1), most (0.6 –0.95), some (0.3 – 0.59)
little (0.06 –0.29), none (0 –0.05) }

and for directional qualifiers is:

{directly (0.96 – 1), mostly (0.6 –0.95), somewhat (0.3 – 0.59),
slightly (0.06 –0.29), not (0 –0.05) }.

The *fuzzy-query* rule in Figure 13 provides the fuzzy querying information by calling *fuzzyTq* and *fuzzyDq* functions.

```
(defrule fuzzy-query
 ?f3 <-(nodes (objectname ?A&A)
            (C ?C_area )
            (N ?N_area ?N_len)
              . . . . . .
            (NW ?NW_area ?NW_len) )
=>
 (if (neq ?A B ) then (bind ?obj B) )
 (loop-for-count (?count 1 8) do
   (bind ?dir (nth$ ?count (create$ North
                          ... North_West)))
   (bind ?area_w (nth$ ?count (create$
                 ?N_area ?NE_area ?E_area
                 . . .?SE_area ?NW_area)))
   (bind ?node_w (nth$ ?count (create$
                 ?N_len ?NE_len ?E_len
                 . . .?W_len ?NW_len)))
   (bind ?tq (fuzzyTq ?A ?area_w
                            ?dir  ?obj))
   (bind ?dq (fuzzyDq ?A ?node_w
                             ?dir ?obj))
   (if (and (neq ?tq non) (neq ?dq non ))
    then
     (printout t “query information” crlf) ) ) )
```

Fig. 13. Fuzzy query rule.

Consider two objects and their respective corner coordinates:

object A (1 , 1) (5 , 3) and
object B (4 , 1)(8 , 7).

When the *define-2D-relation* rule is fired, calling AllenRelation (1 5 4 8) will return ‘o’, and the second calling of AllenRelation (1 3 1 7) will return ‘s.’ Finally, the relation ‘os’ is added to the *spatial-relation* fact.

When the define-topological-relation rule is fired, the topological information 'Object A overlaps Object B' is displayed. When the define-directional-relation rule is fired, 'Object A is South Object B, Object A is South West of Object B, and Object A is West of Object B' are provided for directional relations. When the *reference* rule is fired, the reference object R(4, 1) (5, 3) is asserted into the fact database. While the *get-weight* rule is firing, area weights and node weights are assigned into 9 nodes for each object. Finally, the *fuzzy-query* rule fires, providing the following result:

Most of Object A is West of Object B
Object A is mostly West of Object B
Most of Object A is mostly West of Object B

4.6 Modifications for Anomalous Cases

The original definition of the axis weight of an object sub-group is the ratio of its axis length to the longest axis length of the object sub-groups and it is used to specify how directly an object sub-group lies in a given direction. Under further consideration, the axis weight definition may not directly deal with directions but intuitively is a magnitude ratio between two object sub-groups. In other words, the physical significance of such axis weights regarding to directional information may be somewhat unclear in certain situations. Additionally, when there is no axis going through the MBR, there is no axis weight for this object and so no total node weights can be obtained for the subsequent querying.

Another specific case in which the axis weight definition may not clearly distinguish the directional difference between objects is shown in Figure 14. The reference areas are exactly the same in both Figures 14(a) and 14(b). There is only one object sub-group for each MBR and one and only one axis going through them, therefore the axis weights for both MBRs are 1. However, in Figure 14, object B extends itself markedly to the east. The directional relationships between A and B are apparently different in these two cases. In linguistic terms, in Figure 14(a), object B lies more directly northeast (mostly northeast) of object A, while in Figure 14(b), object B leans much more to the east of A (slightly northeast). The same axis weight values for these two cases, which result in the same total node weights, do not adequately distinguish these intuitively different directional relationships.

Another refinement we have studied is shown in Figure 15, in which objects 2 and 3 are symmetric to axis NE, with 2 on the eastern side, while 3 is to the north. If we calculate axis weights and total node weights for MBR 2 and 3 regarding to MBR 1, the same values will be obtained. To further distinguish the difference in such cases, a parameter termed as close_EW was developed, which specifies whether the node leans to east-west axis or to the north-south axis. It is defined as 1 if the node leans to the east-west axis and 0 if the node leans to the north-south

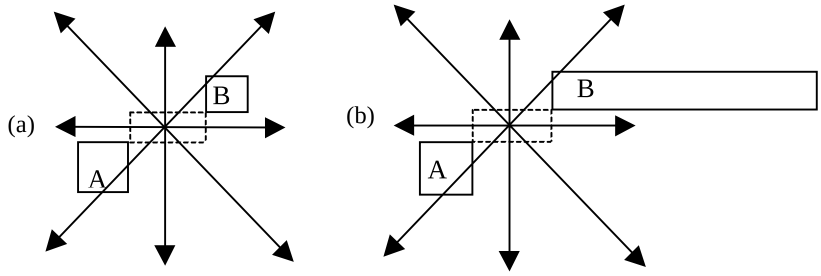


Fig. 14. Distinction between object directional relationships.

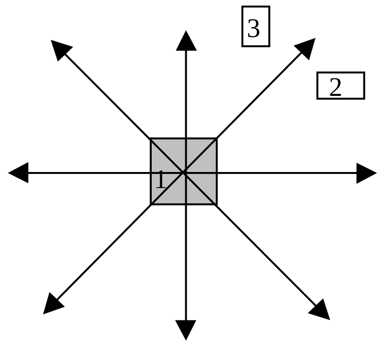


Fig. 15. Direction tendencies of objects.

axis. In Figure 15, close_EW of MBR 2 is 1 and that of MBR 3 is 0, which means that MBR 2 is leaning to the east, while 3 is to the north. In addition to specifying the direction tendencies, close_EW can also be used to further modify the total node weights for querying.

Total node weights can be used to query the directions of the object. Using a threshold on total node weights, as when the total node weight is below 5%, leads us to say the object is not in that direction. Again, somewhat non-intuitive situations can appear in these cases. In Figure 16, MBR 2 has only object sub-group 2 and its total node weight is below 5%. So MBR 2 is considered to be not northeast of MBR 1, but north of MBR 1 in this case. In fact, there is a range of such cases that appear to cause anomalies in an extreme limit situation.

Instead of dealing with all these cases separately when querying, we can formulate the different cases into generalized total node weights and query directions by such weights. From this point forward, the total node weights are intended to mean these generalized total weights, which are derived as follows:

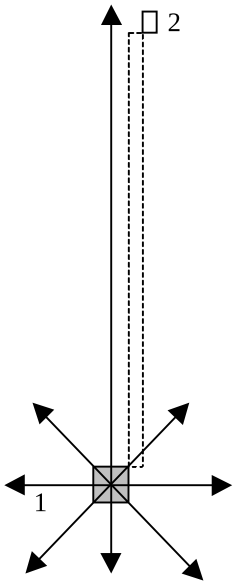


Fig. 16. Object 2 may be considered directly north of object 1.

$(W_{TN})_i$: total node weight of object sub-group i (node i of ASG)
node i: 0 - overlap; 1 - north; 2 - northeast; 3 - east; 4 - southeast
5 - south; 6 - southwest; 7 - west; 8 - northwest

and

$(W_{TN})_0 = (W_{area})_0 * 1$

$(W_{TN})_1 = ((W_{area})_1 + (W_{area})_i) * 1$ i=2,8, if $(W_{node})_i < 0.05$ and $(close\_EW)_i = 0$

$(W_{TN})_i = (W_{node})_i$ i=2,4,6,8 and $(W_{node})_i >= 0.05$

or $(W_{TN})_i = 0$ i=2,4,6,8 and $(W_{node})_i < 0.05$

$(W_{TN})_3 = ((W_{area})_3 + (W_{area})_i) * 1$ i=2,4, if $(W_{node})_i < 0.05$ and $(close\_EW)_i = 1$

$(W_{TN})_5 = ((W_{area})_5 + (W_{area})_i) * 1$ i=4,6, if $(W_{node})_i < 0.05$ and $(close\_EW)_i = 0$

$(W_{TN})_7 = ((W_{area})_7 + (W_{area})_i) * 1$ i=6,8, if $(W_{node})_i < 0.05$ and $(close\_EW)_i = 1$

Now we need to consider only a single total node weight of the corresponding node to perform directional queries. To retrieve all objects that are directly north of the reference object, the querying becomes:

Obtain all objects with $(W_{TN})_1 >= 0.95$.

The total node weights are now generalized total node weights formed by summing up weights for all cases that lead to the same directional information. According to these modified total node weights, directional querying can be conducted by the definitions of directional relationships as shown in Figure 17.

	Directly	Mostly	Slightly	Somewhat	Not
Overlap	$(W_{TN})_0 >= 0.95$	$0.6<=(W_{TN})_0 <0.95$	$0.3<=(W_{TN})_0 <0.6$	$0.05<=(W_{TN})_0 <0.3$	$(W_{TN})_0 <0.05$
North	$(W_{TN})_1 >= 0.95$	$0.6<=(W_{TN})_1 <0.95$	$0.3<=(W_{TN})_1 <0.6$	$0.05<=(W_{TN})_1 <0.3$	$(W_{TN})_1 <0.05$
Northeast	$(W_{TN})_2 >= 0.95$	$0.6<=(W_{TN})_2 <0.95$	$0.3<=(W_{TN})_2 <0.6$	$0.05<=(W_{TN})_2 <0.3$	$(W_{TN})_2 <0.05$
East	$(W_{TN})_3 >= 0.95$	$0.6<=(W_{TN})_3 <0.95$	$0.3<=(W_{TN})_3 <0.6$	$0.05<=(W_{TN})_3 <0.3$	$(W_{TN})_3 <0.05$
Southeast	$(W_{TN})_4 >= 0.95$	$0.6<=(W_{TN})_4 <0.95$	$0.3<=(W_{TN})_4 <0.6$	$0.05<=(W_{TN})_4 <0.3$	$(W_{TN})_4 <0.05$
South	$(W_{TN})_5 >= 0.95$	$0.6<=(W_{TN})_5 <0.95$	$0.3<=(W_{TN})_5 <0.6$	$0.05<=(W_{TN})_5 <0.3$	$(W_{TN})_5 <0.05$
Southwest	$(W_{TN})_6 >= 0.95$	$0.6<=(W_{TN})_6 <0.95$	$0.3<=(W_{TN})_6 <0.6$	$0.05<=(W_{TN})_6 <0.3$	$(W_{TN})_6 <0.05$
West	$(W_{TN})_7 >= 0.95$	$0.6<=(W_{TN})_7 <0.95$	$0.3<=(W_{TN})_7 <0.6$	$0.05<=(W_{TN})_7 <0.3$	$(W_{TN})_7 <0.05$
Northwest	$(W_{TN})_8 >= 0.95$	$0.6<=(W_{TN})_8 <0.95$	$0.3<=(W_{TN})_8 <0.6$	$0.05<=(W_{TN})_8 <0.3$	$(W_{TN})_8 <0.05$

Fig. 17. Modified directional relationship definitions.

4.7 Oracle Implementation

An implementation was performed for the spatial direction approach with Visual C++ 6.0 and Oracle 8i. All calculations, displaying and interfacing were in Visual C++ 6.0. The Oracle 8i database was used to store the coordinates and the ASG weights of each MBR. That database was accessed from Visual C++ by using MFC's ODBC (Open Database Connectivity) database classes. Therefore, the MBR coordinates stored in the database can be obtained to calculate the weights and to display the MBRs, and the calculated weights can be sent to the database for subsequent queries from the programs.

A key issue for the implementation was how to store the total node weights, which are the basis for the directional querying. When directional relationships between two objects are of interest, one of the objects has to be the reference object and if it changes, the directional relationships of the object changes, as well as the total node weights for this object. When there are multiple objects, the issue must be resolved regarding which object to use as the reference for calculating the weights for all the other objects. These weights are then stored in the database. Another option is to pre-calculate each object's total node weights using every other object as a reference. This results in a set of total node weights being stored for each object, rather than a single weight. In this way, the total node weights between any combination of objects are calculated, eliminating the need for directional transitivity computations when querying. However, a huge amount of storage is required for this method, thus making it largely impractical for all but the smallest of datasets.

To resolve this problem, a dynamic database was utilized. A static table stores the coordinates of each MBR, and another table is used to store the total node weights for those MBRs dynamically. Initially, the total node weight table stores no information. Every time a query needs to be conducted, the reference MBR is set first and the total node weights are calculated accordingly and stored in the table, and the query is conducted upon this table. Any subsequent query based on the same reference MBR can be performed on this table without recalculation of the weights. If the reference MBR for the subsequent query changes, the stored information in the total node weight table is removed first, then the reference MBR for computation is reset, followed by recalculation and storage of the total node weights for all MBRs and the querying. By calculating and storing the total node weights dynamically, no directional transitivity needs to be considered, and the amount of information that needs to be stored is small. This method has also proven to provide satisfactory querying.

Directional Querying

Two types of directional querying were implemented in this work. The first one queries the directional relationships between any two MBRs (one of them is the reference MBR), such as "Get the directions of MBR A to MBR B (B is the reference MBR)." The other type of querying obtains all MBRs that lie in a specific direction of the reference MBR, for example, "Retrieve all objects that are directly northwest of object A."

When querying is on the directional relationships between two MBRs, only the total node weights of the queried MBR are calculated. The weights are not stored into the total node weight table of the database, but translated into linguistic terms like "directly north," "mostly southwest" or "somewhat south," and displayed to the user.

When the querying is to retrieve all MBRs in a specific direction from the reference MBR, total node weights for all MBRs except the reference MBR must be calculated. The weights are then stored into the total node weight table of the database and can be accessed by using the MFC's ODBC CRecordset class, DirectionWeightsDB. The querying is then realized by executing some SQL statements on the DirectionWeightsDB object.

We will now describe the first option as an example—retrieve the directional relationships between two objects. At first, the working MBR, for which directions are to be queried, must be set. Any existing MBR number can be entered in the dialog in Figure 18. In case of an invalid value, there is a prompt to re-enter a valid number. After setting the working MBR, its directions upon the reference MBR can be obtained by clicking on the "Get Directions" button and the directions will be displayed in the dialog box. For example, the working MBR in Figure 18 is MBR 3, its directions to MBR 0 are:

"Mostly east, somewhat north, somewhat south, and somewhat overlapping."

In addition, if the "Get Picture" is clicked, the working MBR will be re-displayed in red. Therefore, the two MBRs, the working MBR and the reference MBR, will stand out from the others for ease of evident visualization of their directional relationships.

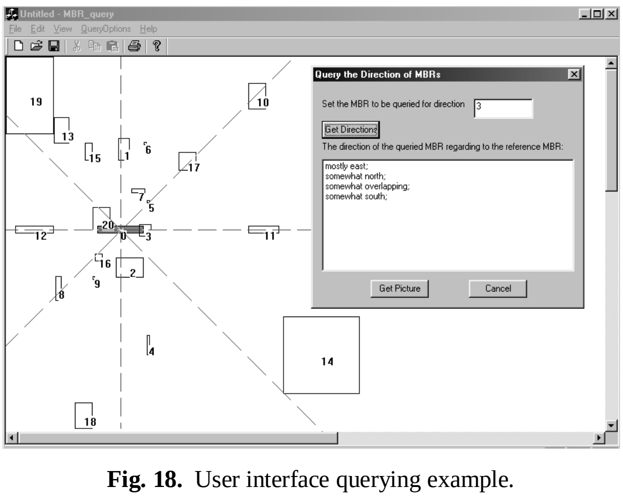


Fig. 18. User interface querying example.

5 Intelligent Agent Technology

In this section, we provide a brief introduction to the topic of intelligent agents, and present an agent-based framework for implementing the ASG model within a larger, distributed system.

5.1 Overview

We can define an agent as anyone or anything that acts as a representative for another party, for the express purpose of performing specific acts that are seen to be beneficial to the represented party. A software agent, which has been around for approximately a decade, is a software program that performs tasks for its user within a computing environment. Technically speaking, most fourth generation application software could be defined as agents. Every day we ask computers, through software, to perform hundreds of different tasks for us, essentially calling upon their agency attributes.

As we descend deeper into the concept of agency, we can see that there are distinct characteristics that collectively constitute an intelligent software agent. Intelligent software agents, or intelligent agents for short, are differentiated from other applications by their added dimensions of mobility, autonomy, and the ability to interact independent of its user's presence. When we introduce the additional element of intelligence to an agent, we must include the ability for adaptive reasoning. This implies the capability to process information from external environments — such as networks, databases, and the Internet — given a set of knowledge, attitudes, and beliefs of the user as understood by the agent.

There are six key fundamental characteristics of intelligent agents that differentiate them from other types of software applications [26]:

- Autonomy,
- Communication Ability,
- Capacity for Cooperation,
- Adaptive Behavior,
- Trustworthiness, and
- Capacity for Reasoning and Learning.

The ability to perform reasoning and learning is one of the key aspects of intelligence that distinguishes intelligent agents from other more "robotic" agents. As Belgrave [6] describes, reasoning implies that "an agent can possess the ability to infer and extrapolate based on current knowledge and experiences - in a rational, reproducible way." Based on our investigation there are five types of reasoning and learning scenarios:

- Rule-based reasoning, where agents use a set of user pre-conditions to evaluate conditions in the external environment,
- Knowledge-based reasoning, where agents are provided large sets of data about prior scenarios and resulting actions, from which they deduce their future moves,
- Simple Statistical Analysis for learning and reasoning,
- Fuzzy Agents for reasoning when the information is imprecise or incomplete,
- Neural Networks reasoning for unstructured data or noisy data, and
- Evolutionary Computing to expand the learning by viewing the system from an inter-agent perspective and employing a generic algorithm.

5.2 Rule-Based Reasoning

Of all the technologies used to build intelligent agents, the easiest to understand is rule-based reasoning, the basis for "inference engines." Agents use the set of rules to decide which action or actions they should take ("If a condition C is satisfied, then perform action A"). With multiple rules, one rule's action may cause the satisfaction of another rule's conditions. This kind of chained effect is called forward chaining, and the problem with this approach is that the user needs to recognize the opportunity for employing an agent, take the initiative in programming the rules, endow the agent with explicit knowledge specified in an abstract language, and maintain the rules over time, as habits or events change.

IBM's RAISE (Reusable Agent Intelligence Software Environment) is an example of rule-based reasoning. RAISE is the inference engine of IBM's Agent Building Environment (ABE) developer's toolkit. It can perform information flow functions: finding, searching, filtering, categorizing, storing, routing, and/or selectively disseminating information items. Prototype applications for RAISE include e-commerce shopping, customer service support, and workflow on the Web and in Lotus Notes, news, and e-mail [26].

MAGSY multi-agent rule-based system is another example. The Kernel of an agent in MAGSY is a forward-chaining rule interpreter; therefore, each agent has the problem solving capacity of an expert system. In this kind of multi-agent system, the knowledge of the agents is usually structured in an object-oriented knowledge representation scheme. There is a global knowledge base, which contains the knowledge that may be accessed by all of the agents. Agents may store their identification in this global knowledge base and thus become known to all agents in the system [26].

One problem with rule-based systems is that users must keep them updated manually. These systems cannot change by themselves. A second problem is that complex sets of rules may develop conflicting rules that the agent can't resolve.

5.3 Knowledge-Based Reasoning

Knowledge-based systems (KBS) are a relatively mature aspect of artificial intelligence technology. These systems solve problems in complex application and ill-structured domains by using a large body of explicitly represented domain knowledge to search for solutions.

One can build knowledge bases based on a specific subject area or domain. These then serve as the basis for some agent inference mechanisms, including the rule-based reasoning techniques mentioned above. Usually, the developers or the knowledge engineers endow programs with information about the tasks to be performed by an agent in a specific domain, and the agent infers the proper response to a given situation. The major problem with such systems is that they require a large amount of work from the knowledge engineers. Furthermore, the knowledge of the agent is fixed and cannot be customized to the desire of individual users. In highly personalized applications, the knowledge engineer cannot possibly anticipate the best aid for each user in each situation.

In order to solve certain kinds of complex problems by knowledge-based agents, it is beneficial to create a system in which a number of Knowledge Base Agents (KBAs) cooperate and combine their problem solving capabilities. Sometimes this occurs because the problem-solving activity covers a large geographic region (such as in telecommunication networks and military applications), where different KBAs have responsibility for different geographical areas. Sometimes it occurs because different KBAs have different “specialties” to bring to the problem-solving process, similar to the co-operation among human team members. The Multiple-Agent System (MAS) paradigm has proven a popular and effective method for building a co-operating team of KBA. Each KBA in the team is constructed as a software agent, conferring abilities of autonomy, self-knowledge, and acquaintance knowledge on the KBA abilities useful for team-forming and co-operative problem solving.

5.4 Implementation

The general objective of our work is to develop an autonomous multi-agent system that retrieves, filters, integrates/conflates and validates geo-spatial data from multiple heterogeneous sources. It is assumed that the data can be stored in a number of formats, the geo-spatial databases can be relational as well as object-oriented, and different software vendors can be the sources of the database environments. Finally, the data sources are distributed and include selected proprietary databases as well as, potentially, web-based resources.

Because of the distributed and complex nature of the problem, agent technology was selected as the implementation paradigm. Intelligent agents offer several advantages over standard client-server architecture. The ability to move processing to the source of activity, i.e., a database server, reduces the network overhead. Moreover, multiple agents can simultaneously process information stored in multiple data locations. Such agents can communicate and cross-reference distributed data to support the distributed conflation process. The ability to deploy software to a remote site in an unobtrusive manner extends the functionality of the local server and allows for utilization of additional available resources. Finally, as soon as agents are deployed out of the local machine, the connection to the network can be closed (e.g., supporting the security of the system), or the local host can even be shutdown without adversely affecting the data acquisition process. A system diagram is shown in Figure 19.

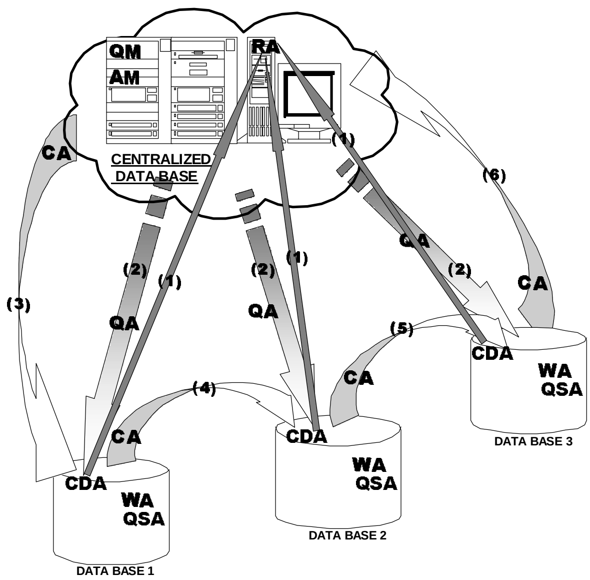


Fig. 19. Mobile agent system for distributed spatial databases.

The work presented in this chapter is implemented in the ConflationAgents (CAs). The CA is a superclass of many specialized agent classes that have extensive knowledge about their domain relevant to the conflation process. CAs are intelligent mobile agents, traveling to the feeder databases to perform conflation. The CA traverses all of the relevant databases, collecting the information and executing the knowledge-based conflation algorithm, of which fuzzy spatial relationship matching is a component. For more details on the mobile agent system interactions, please refer to [26]. As of now, a primitive implementation of the CA based on the Java Expert System Shell (JESS) has been initiated.

6 Summary and Future Work

This chapter has provided a discussion of several issues related to uncertainty in spatial relationships. We presented a brief summary of others' approaches to the topic, followed by a somewhat in-depth definition of our model based on ASGs. We showed how this model supports the definition of both fuzzy directional and fuzzy topological relationships. Several modifications of the original model were also presented, including one based on MBR refinements, one to support a specific implementation and one to handle certain cases that can produce anomalous query results. Two implementations, one for an Oracle database environment and the other based on an expert system shell were also described. We also discussed the applicability of mobile agent technology to implementation of the model for a distributed system.

Our next step is to fully implement the ASG model in a mobile agent environment. We plan to provide a prototype system that supports multiple, heterogeneous spatial databases. Fuzzy querying of spatial relationships is an essential component both for user interaction and for more complex tasks related to spatial data conflation. We believe that the ASG model or one of its variants will provide a reliable framework for a robust, working system.

Acknowledgments

We would like to thank the National Imagery and Mapping Agency, the Marine Corps Warfighting Lab, PE 0603640M, the Office of Naval Research, PE 0603238N, and the Naval Research Laboratory for sponsoring this research.

References

1. J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832-843, 1983.
2. J. Baldwin. Knowledge Engineering Using a Fuzzy Relational Inference Language. In *IFAC Symposium on Fuzzy Information Knowledge Representation and Decision Analysis*, pages 15-21, 1983.
3. B. Buckles and F. Petry. A Fuzzy Model for Relational Databases. *Fuzzy Sets and Systems*, 7:213-226, 1982.
4. P. Burrough and A. Frank. *Geographical Objects and Indeterminate Boundaries,* Taylor and Francis, London UK, 1996.
5. P. Burrough, P. van Gaans, and R. MacMillian. High-resolution landform classif-ication using fuzzy k-means. *Fuzzy Sets and Systems*, 113(1):37-52, 2000.

6. M. Belgrave. The Unified Agent Architecture: A White Paper. *URL: http://www.ee.mcgill.ca/ ~belmarc/uaa_paper.html*, 1999.
7. C. Chang and T. Wu. Retrieving the Most Similar Symbolic Pictures from Pictorial Databases. *Information Processing and Management*, 28(5):581-588, 1992.
8. E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling Topological and Spatial Relations: Strategies for Query Processing. *Computers and Graphics*, 18(6):815-822, 1994.
9. *CLIPS Reference Manual*, Version 6.10, August 5th, 1998.
10. M. Cobb. *An Approach for the Definition, Representation and Querying of Binary Topological and Directional Relationships between Two-Dimensional Objects*, Ph. D. Thesis, Tulane University, 1995.
11. M. Cobb and F. Petry. Modeling Spatial Data within a Fuzzy Framework. *Journal of Amer. Soc. Information Science*, 49:253-266, 1998.
12. M. Cobb, F. Petry, and V. Robinson. Special Issue: Uncertainty in Geographic Information Systems and Spatial Data. *Fuzzy Sets and Systems*, 113(1), 2000.
13. V. Cross and A. Firat. Fuzzy objects for geographical information systems. *Fuzzy Sets and Systems*, 113(1):19-36, 2000.
14. S. Dutta. Topological Constraints: A Representational Framework for Approximate Spatial and Temporal Reasoning. In *SSD'91 (Advances in Spatial Databases: 2nd Symposium)*, pages 161-180, 1991.
15. M. J. Egenhofer and R. D. Franzosa. Point-set Topological Spatial Relations. *Int. Journal of Geographical Information Systems*, 5(2):161-174, 1991.
16. Environmental Systems Research Institute (ESRI). *ARC/INFO® User's Guide: ARC /INFO® 6.0 Data Model, Concepts and Key Terms*, Environmental Systems Research Institute, Redlands, CA, 1992.
17. A. Galton. *Perturbation and Dominance in the Qualitative Representation of Continuous State-Spaces*, Technical Report 27O, Department of Computer Science, University of Exeter, Exeter, 1994.
18. C. Giardina. *Fuzzy Databases and Fuzzy Relational Associative Processors*, Technical Report, Stevens Institute of Technology, Hoboken NJ, 1979.
19. M. Goodchild and S. Gopal (Eds.). *The Accuracy of Spatial Databases*, Taylor and Francis, Basingstoke, UK, 1990.
20. H. W. Guesgen. *Spatial Reasoning Based on Allen's Temporal Logic*, Technical Report TR-89-049, International Computer Science Institute, Berkeley, CA, 1989.
21. H. P. Kriegel, M. Schiwietz, R. Schneider, and B. Seeger. Performance comparison of Point and Spatial Access Methods. In: A. Buchmann, O. Gunther, T. Smith, and Y. Wang (Eds.), *Design and Implementation of Large Spatial Databases*, LNCS 409, pages 89-114, Santa Barbara, CA, Springer-Verlag, 1989.
22. D. MaGuire. An Overview and Definition of GIS. *Geographical Information Systems: Principles and Applications*, D. MaGuire, M. Goodchild, and D. Rhind (Eds.), 1:9-2O, Longman, Essex GB, 1991.
23. A. Mukerjee and G. Joe. A Qualitative Model for Space. In *AAAI-90 (8th National Conference on Artificial Intelligence)*, pages 721-727, 1990.
24. M. Nabil, J. Shepherd, and A. H. H. Ngu. 2D Projection Interval Relationships: A Symbolic Representation of Spatial Relationships. In *SSD '95 (Advances in Spatial Databases: 42nd Symposium)*, pages 292-309, 1995.
25. D. Papadias and Y. Theodoridis. Spatial Relations, Minimum Bounding Rrectangles, and Spatial Data Dtructures. *Int. J. Geographical Information Science*, 11:111-138, 1997.
26. S. Rahimi, A. Ali, and D. Ali. An Investigation on Intelligent Software-Agent Technology. In *IEMS and IC&IE Joined Int. Conference*, Cocoa Beach, Florida, 2001.
27. V. Robinson. Implications of Fuzzy Set Theory for Geographic Databases. *Computers, Environment, and Urban Svstems*, 12:89-98, 1988.

28. V. Robinson. Interactive Machine Acquisition of a Fuzzy Spatial Relation. *Computers and Geosciences*, 6:857-872, 1990.
29. V. Robinson and A. Frank. About Different Kinds of Uncertainty in Geographic Information Systems. In *AUTOCARTO 7*, 1985.
30. H. Samet. Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS. Reading, MA: Addison-Wesley, 1989.
31. J. Sharma and D. M. Flewelling. Inferences from Combined Knowledge about Topology and Direction. In *SSD'95 (Advances in Spatial Databases: 42nd Symposium)*, pages 279-291, 1995.
32. D. Stoms. Reasoning with Uncertainty in Intelligent Geographic Information Systems. In *GIS'87 (2nd Annual Int. Conf on Geographic Information Systems)*, pages 693-699, American Soc. for Photogrammetry and Remote Sensing, Falls Church VA, 1987.
33. F. Wang. A fuzzy grammar and possibility theory-based natural language user interface for spatial queries. *Fuzzy Sets and Systems*, 113(1):147-159, 2000.
34. S. Winter. Topological Relations between Discrete Regions. In *SSD'95 (Advances in Spatial Databases: 42nd Symposium)*, pages 310-327, Portland, ME, USA, August 1995.
35. S. Winter. Uncertainty of Topological Relations in GIS. In *ISPRS Commission III Symposium*, pages 924-930, 1994.
36. R. M. Wygant. CLIPS – A Powerful Development and Delivery Expert System Tool. *Computers in Engineering*, 17(1/4):546-549, 1989.
37. L. Zadeh. Test-Score Semantics for Natural Languages and Meaning Representation via PRUF. *Empirical Semantics*, B. Rieger (Ed.), pages 281-349, Brockmeyer, Bochum, GR, 1981.