Relation-Based Similarity

Dimitris Papadias  
Department of Computer Science  
Hong Kong University of Science and Technology  
Clear Water Bay, Kowloon, Hong Kong  
dimitris@cs.ust.hk  

Vasilis Delis  
Computer Engineering and Informatics Dept.  
University of Patras  
and Computer Technology Institute  
Kolokotroni 3, 26221, Patras, Greece  
delis@cti.gr

Abstract: Similarity queries constitute an active area in spatial query processing. The paper addresses the problem of qualitative similarity based on spatial relations. On the one hand spatial similarity entails mechanisms for representing and reasoning on spatial relations, while on the other it introduces a high level of uncertainty. Several spatial queries can be rather fuzzy and user or application dependent. Moreover, relations such as near, northeast etc. lack universally accepted semantics, and as a result their processing in Spatial Databases and GIS has to provide a high level of flexibility in order to satisfy real-life needs. Our work extends the notion of conceptual neighbourhood (originally defined for 1D space) to higher dimensions and proposes a unified multiresolution framework for the handling of topological, directional and distance relations. We discuss how object and image similarity queries can be effectively handled and how uncertainty can be seamlessly incorporated in our model.

1. INTRODUCTION  
This paper studies similarity retrieval based on spatial relations. We assume a database of images (or maps) of distinct objects with certain properties, and queries that retrieve objects and images that satisfy some either well-defined or fuzzy characteristics, the latter being more interesting and usually more difficult to handle: e.g., “find all industrial zones near a lake in area A” or “find all images similar to image I”. The expected output of both queries is not well-defined, but depends on the particular user’s notion of near and similar. This work uses relation similarity in order to model this type of uncertainty. Relation-based similarity is qualitative, because it does not rely on absolute coordinates and actual shapes, but on relative objects’ positions.

The spatial relations that we assume in this work are projection-based extensions (Papadias and Egenhofer, 1997) of Allen’s (1983) work on (convex and closed) temporal intervals. Freksa (1992) defined the conceptual neighbourhood for Allen’s 13 primitive relations as shown in Figure 1. If we start with relation $R_1$ and extend (or move) the upper interval to the right we derive relation $R_2$. With a similar extension we can derive the transition from $R_2$ to $R_3$ and so on. $R_2$ and $R_3$ are called 1$^{st}$ degree neighbours of $R_2$. Previous work on spatial relation similarity has been carried out for topological relations (Egenhofer and Al-Taha, 1992) and for classes of both topological and direction relations (Bruns and Egenhofer, 1996).
3. RELATION-BASED OBJECT RETRIEVAL

An important class of queries in Spatial Databases and GIS consists of queries of the form: "find all (primary) objects that satisfy a given relation set \( r \) with respect to a given reference object \( X \)." The problem with such queries is that usually spatial relations cannot be modelled by boolean domains, but the difference between objects that satisfy the query, and the ones that don't, may be quantifiable and gradual. Information retrieval techniques (like WWW search engines), deal with this problem by associating the retrieved documents with a score proportional to the similarity of the query and the document (Salton et al., 1994). In an analogous manner, the output objects in the case of spatial queries should have an associated "score" to indicate the similarity between their relation and the target relations of the query relation set. Here the score is inversely proportional to the degree of neighbourhood (distance in the graph).

To illustrate the above ideas, we consider a situation of a flooded city where the mass of water moves abruptly and a GIS for damage assessment. The database contains images (Figure 4) describing instances of the flood at different times, and the queries are of the form "find all residential areas covered by the flood at any time (in any image)". Figure 4 also includes a simple object-class hierarchy. Objects in queries can be specific instances (e.g., flood) or generic classes (residential area) or disjunctions thereof (residential area or commercial center).

There are two types of uncertainty associated with object retrieval. The first one is due to the inherent fuzziness of some spatial relations. Assume that \( \text{Northeast} \), is mapped onto the relation set \( \{R_{1,13}\} \); all objects that are related with \( R_{1,13} \) with respect to a reference object \( X \) are said to be \( \text{northeast} \) of \( X \). However, additional neighbouring relations (such as \( R_{1,12}, R_{2,13} \)) may also be regarded as \( \text{northeast} \), depending on the shape and the size of the object, or the user's expectations. The corresponding objects should be retrieved (with a lower score) and the user should decide about their relevance to the query.

The second type of uncertainty is due to fuzzy boundaries. Several spatial objects do not have well defined boundaries (e.g., residential areas, forests). Others have boundaries that change...
over time (shorelines in the presence of tide). In such cases the stored objects are only approximations of actual ones (they can be slightly larger or smaller), and as a consequence, the derived projection relations may be inaccurate. Conceptual neighbourhoods can deal with both forms of uncertainty.

Formally an object retrieval query \( Q_0 \) is a 5-tuple \((C_Y, r, X, i, d)\). The result of the query consists of the retrieved object instances which belong to the class \( C_Y \) and satisfy the relation set \( r \) with respect to the object class or instance \( X \), in a set of images \( i \), at a maximum tolerance (relation distance) \( d \). Let \( x \) be a retrieved object and \( X_I \) an instantiation of \( X \) (unless \( X \) is already an instance in which case \( X_I = X \)). The score of \( x \) is \( \text{MAX}\_\text{Degree}\_\text{of}\_\text{Neighbourhood} \) (which is the maximum distance in the neighbourhood graph and equals to \( \delta \) in 1D and \( \delta \) in 2D assuming Allen’s relations) minus the minimum distance between \( R_X(x, x) \) and the relation set \( r \). The following pseudo-code implements the above described object retrieval. 

\[ \text{Find Instantiations}() \] function returns the domain of possible instantiations of the reference object in an image.

```c
Object_Retrieval(Generic_Primary_Object C_Y, Relation_Set r, Reference_Object X, Image_Set i, real Degree) {
  Object_Set result = {}; 
  for each object Y: score(Y) = 0;
  for each image I in i {
    for each reference object X, Image_Set i, real Degree) {
      Domain(I, X) = \{X\};
      for each primitive relation Rkl in r {
        used(Rkl) = false; counter = 0;
        do {
          for each primitive relation Rkl in r such that not(used(Rkl)) {
            for each object Y such that Rkl(y, a) in I and C_Y = superclass(Y) {
              current_score = \text{MAX}\_\text{Degree}\_\text{of}\_\text{Neighbourhood} - counter;
              if (current_score > score(Y)) score(Y) = current_score;
              result = result ∪ Y;
              r = \text{current_score} ∪ \text{Neighbours}(Rkl);
              used(Rkl) = true;
            }
            counter++;
            while (counter >= Degree);
          } // end-for reference objects
        } // end-for images
        return result;
      } // Find Instantiations(Image I, Object X)
      if (X is object class) {
        let C_x be the subclass(es) of X,
        Domain(LX) = \{A / A ∈ 1 and A ∈ C_x\};
        } // X is instance
      if (X ∈ I) Domain(LX) = \{X\};
      else Domain(LX) = \{X\}; // empty Instantiation.
    } return Domain(LX) // returns the possible instantiations of X in image I
  } // end-for images
}
```

Figure 5 illustrates the processing of the query "Find all residential areas or commercial centers northeast of the flood in any image with a target degree 2" (Object_Retrieval (Residential Areas, Commercial Centers, \( R_{1-13} \), Flood, \( Y \_1, Y \_2 \), 2)). Initially image \( I_1 \) is processed and Domain(I, Flood) = \{Flood\}, \( r = (R_{1-13}) \). \( C_1 \) is retrieved because the relation \( R_{1-13}(C_1, \text{Flood}) \) holds, with a score equal to \( \text{MAX}\_\text{Degree}\_\text{of}\_\text{Neighbourhood} \). The neighbours of \( R_{1-13} \) \( (R_{1-12}, R_{2-13}) \) are then added to \( r \), and \( R_{1-13} \) becomes used, in order to avoid retrieving object \( C_1 \) again (used relations are denoted with italics in Figure 5). During the second iteration of the do-loop (counter=1) \( C_2 \) is retrieved, because \( R_{2-13} \) \( (C_2, \text{Flood}) \). The last object to be retrieved from image \( I_1 \) is \( A_2 \) with a score of 14. Then \( I_2 \) is processed and the same procedure is followed.

<table>
<thead>
<tr>
<th>Relation Set</th>
<th>Primary Object</th>
<th>Relation</th>
<th>Score</th>
<th>Current Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>( R_{1-13} )</td>
<td>( C_1 )</td>
<td>( R_{1-13} )</td>
<td>16</td>
</tr>
<tr>
<td>( I_1 )</td>
<td>( R_{1-13}, R_{1-12}, R_{2-13} )</td>
<td>( C_2 )</td>
<td>( R_{2-13} )</td>
<td>15</td>
</tr>
<tr>
<td>( I_1 )</td>
<td>( R_{1-13}, R_{1-12}, R_{2-13}, R_{1-11}, R_{3-13} )</td>
<td>( A_2 )</td>
<td>( R_{1-11} )</td>
<td>14</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>( R_{1-13} )</td>
<td>( C_1 )</td>
<td>( R_{1-13} )</td>
<td>16</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>( R_{1-13}, R_{1-12}, R_{2-13} )</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>( R_{1-13}, R_{1-12}, R_{2-13}, R_{1-11}, R_{3-13} )</td>
<td>( C_2 )</td>
<td>( R_{2-13} )</td>
<td>15</td>
</tr>
</tbody>
</table>

**Figure 5 Example query processing**

The above example does not represent the most general Object_Retrieval query, since it involves only one (instance) reference object. In case of a reference object class, we would have to perform the same processing for each instantiation of the reference object, in every image. The cost \( C \) of retrieval of all objects that satisfy a relation set in an image, depends on the relation and the underlying data structure. This cost is measured by the number of disk accesses and, in general, increases with the area covered by the projection relation. The previous algorithm calls the spatial data structure a number of times equal to \( m = \text{Number of Images} \times \text{Number of Instantiations in Each Image} \times \text{Degree} \), resulting in a total cost of \( m^4 \).

Unlike the fuzziness of direction relations, for topological relations there exists a set of formal and widely accepted definitions based on the intersection model (Egenhofer and Franzosa, 1991). Although, assuming the intersection model, the expected output for topological queries is unambiguous, conceptual neighbourhoods are still useful to deal with fuzzy boundaries. Consider the query: "find all residential covered by the flood in any image". The MBRs that contain potential answers to the query (in the ideal situation) are the ones covered_by, inside, or equal to the MBR of flood. In the presence of fuzzy boundaries, however, we need to retrieve some additional MBRs that satisfy neighbouring relations, in order to make sure that we don’t miss any objects (for details see Papadias et al., 1995). All queries involving combinations of spatial relations with respect to a reference object can be derived by proper application of Object_Retrieval. In case of disjunctions of spatial relationships, the relation set consists of the union of the relation sets for each relationship. For conjunctions (e.g., northeast and covered by), the input of the primitive query is the intersection of the individual sets.

4. RELATION-BASED IMAGE RETRIEVAL

Another important class of queries retrieves images that satisfy a set of given relations between distinct objects. e.g., "find all images where some residential area southwest of a commercial center is covered_by the flood". Our interest is not to give a binary answer (yes, no) when assessing an image. Rather, a more interesting (and useful in practice) attempt would be to rank all images according to their resemblance to the queried spatial configuration. Under this perspective, the above query is essentially an image similarity query, since it describes a generic query-image (depicting some residential area southwest of a commercial center, covered by the flood) which is matched with the stored images in order to retrieve the most similar ones. Similarity is only based on the properties of the objects to be matched (e.g., residential area,…) and their interrelationships but not on their visual characteristics (e.g. shape, size).

Such queries can be formalised as finite sets consisting of 3-tuples of the form \((X, Y, r)\), where \(X\) and \(Y\) are object classes or instances and \(r\) is a set of primitive projection relations which
semantically corresponds to their disjunction. Each pair \((X, Y)\) must satisfy the corresponding relation set \(r\). For example, the query expressed in the previous paragraph can be formally expressed as: \(Q_t = \{(\text{residential area, commercial center,} \{\text{southwest}\}), \text{\{residential area, flood, \{covered_by\}}\}\)

During the execution of the query, stored images are sequentially examined and different instantiations of pairs of objects are assessed for matching each of the above tuples. Consider a generic query \(Q = \{(X_i, Y_i, R_i) \mid i = 1..n\}\) and a particular image instantiation \(I = \{(X_{I_i}, Y_{I_i}, R_{I_i}) \mid i = 1..n\}\). The similarity measure that we adopt is:

\[
\text{Similarity}(I, Q) = \sum_{i=1}^{n} \left( \text{Max Degree of Neighbourhood} - d(R_i, r_i) \right)
\]

The maximum similarity is equal to \(\text{MAX Degree of Neighbourhood}\) when images contain exactly the same query objects related by the same binary projection relations. The minimum similarity is 0 when none of the query objects can be instantiated, or in the extreme case where the distance between all instantiated relations and the corresponding query relations is \(\text{MAX Degree of Neighbourhood}\). We will illustrate image similarity using Figure 4, assuming one of the images to be the query image (\(n=10\). The resultant similarity measure is 15.5 (the only differences between the images are caused by the movement of the flood).

<table>
<thead>
<tr>
<th>Image I₁</th>
<th>Image I₂</th>
<th>degree of neighbourhood</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁₂₁(A₁, A₂)</td>
<td>R₁₂₁(A₁, A₂)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₁₃₁(A₁, C₁)</td>
<td>R₁₃₁(A₁, C₁)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₁₁₁(C₁, C₂)</td>
<td>R₁₁₁(C₁, C₂)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₃₈₈(A₁, Flood)</td>
<td>R₃₈₈(A₁, Flood)</td>
<td>(d = 2)</td>
</tr>
<tr>
<td>R₃₃₃(A₂, C₁)</td>
<td>R₃₃₃(A₂, C₁)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₃₃₃(A₂, C₂)</td>
<td>R₃₃₃(A₂, C₂)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₁₁₁₁(A₂, Flood)</td>
<td>R₁₁₁₁(A₂, Flood)</td>
<td>(d = 2)</td>
</tr>
<tr>
<td>R₁₃₁₃(C₁, C₂)</td>
<td>R₁₃₁₃(C₁, C₂)</td>
<td>(d = 0)</td>
</tr>
<tr>
<td>R₂₂₂₂(C₂, Flood)</td>
<td>R₂₂₂₂(C₂, Flood)</td>
<td>(d = 1)</td>
</tr>
</tbody>
</table>

\[\text{Similarity}(I₁, I₂) = \frac{16*10 - (2 + 2 + 1)}{10} = 15.5\]

**Figure 6** An example similarity assessment

Nabil et al. (1996) also propose a projection-based technique that uses conceptual neighbourhoods to measure image similarity. However they only deal with cases where images are re-arrangements of the same set of objects. The queries simply retrieve all images where some configuration of specific objects is satisfied (e.g., "find all images where object A is above object B..."). The general problem of configuration similarity where images contain arbitrary objects and the queries refer to object variables rather than instances is exponential to the number of objects in stored images because of possible multiple instantiations.

Applying the query \(\{(\text{residential area, commercial center,} \{\text{southwest}\}), \text{\{residential area, flood, \{covered_by\}}\}\} \) to the images of Figure 4 we get two possible instantiations for the residential area object, and two instantiations for the commercial center object, resulting in a total of four instantiations for each image (Figure 7). Every instantiation produces a different sub-image with its own similarity. The user may impose a degree of acceptability, so the result consists of all subimages that have a difference from the target score less than a given degree. In the example of Figure 7, if the given degree is 1, the first sub-image of I₁, and the two first sub-images of I₂ will be returned to the user.

**Figure 7** Example image similarity retrieval

The exponential structure of image similarity retrieval is problematic for applications involving large images. However in many situations a large number of instantiations produces differences larger than the target early in the search process. In this case we do not need to proceed with the rest of the objects, because even if their instantiations result in perfect relation matches, the target difference cannot be satisfied.

5. CONCLUSION

In this paper we discuss a form of spatial similarity based on relations. First, we introduce a generalised framework for the definition of projection-based relations in N-dimensional spaces. Topological, directional and qualitative distance relations can be expressed in various granularities and reasoning on conceptual neighbourhoods is significantly facilitated. On these grounds we develop a framework for handling object and image similarity queries which can manage uncertainty in the definition of spatial relations and fuzzy spatial objects’ boundaries. Our results are important for most domains involving spatial data, such as GIS and CAD.

**REFERENCES**


Bruns, T.H., Egenhofer, M.J. “Similarity of Spatial Scenes”. 7th Symposium on Spatial Data Handling (SDH), 1996.


