Handling Multiple Instances of Symbols in Pictorial Queries by Image Similarity

Aya Soffer[1] and Hanan Samet[2]

[1] CESDIS, Goddard Space Flight Center and
Center for Automation Research, University of Maryland at College Park and
Computer Science and EE Department, University of Maryland Baltimore County
5401 Wilkens Avenue, Baltimore, MD 21228-5398
E-mail: soffer@cs.umbc.edu ***

[2] Computer Science Department and,
Center for Automation Research and Institute for Advanced Computer Science
University of Maryland at College Park, College Park, Maryland 20742
E-mail: hjs@umiacs.umd.edu †

Abstract. A method is presented for processing pictorial query specifications that consist of a query image and a similarity level that must hold between the query image and database images. The similarity level specifies the *contextual similarity* (how well does the content of one image match that of another) as well as the *spatial similarity* (the relative locations of the matching symbols in the two images). This method allows more than one instance of each object in the database image (while still requiring only one instance of each object in the query image). The algorithm tries to satisfy the contextual similarity first and then tries to satisfy the spatial constraints using an auxiliary graph data structure. The running time of this method is exponential in the number of objects in the query image.

1 Introduction

A basic requirement of an image database is the ability to query the database pictorially. The most common method of doing this is querying via an example image. The problem with this method is that in an image database we are usually not looking for an exact match. Instead, the goal is to find images that are similar to a given query image. The main issue is how to determine if two images are similar and whether the similarity criteria that are used by the database system match the user's notion of similarity. There are a few commercial systems that support retrieval of images by pictorial specification [6, 12]. Numerous prototype research IDMS's deal with storage and retrieval of images [1, 3, 4, 7, 8, 11].

In [9] we introduced a method for specifying queries to an image database pictorially that enables the user to indicate the level of required similarity between

*** The support of USRA/CESDIS and NASA Goddard Space Flight Center is gratefully acknowledged.

† The support of the National Science Foundation under Grant IRI-92-16970 is gratefully acknowledged.

the query image and the database images. This method of query specification and the corresponding query processing techniques are applicable to images where the set of objects that may appear in them is known a priori, and where these objects are represented by graphical symbols. For example, in the map domain, graphical symbols are used to indicate the location of various sites such as hospitals, post offices, recreation areas, scenic areas etc. We call this class of images *symbolic images*. The query image is constructed by positioning objects so that the desired spatial constraints hold. Two image similarity factors are considered: (1) *contextual similarity*: how well does the content of one image match that of another. For example, should the database image contain all of the objects in the query image or may it just contain some of these objects. (2) *spatial similarity*: the relative locations of the matching symbols in the two images. By specifying the desired contextual and spatial similarity levels along with the query image, users can specify the extent of the required similarity.

The query processing algorithms in [9] assume only one instance of each symbol in the query image as well as in the database images. In this paper we present a method that allows the occurrence of more than one instance of each symbol in the database image (while still requiring only one instance of each symbol in the query image). This method tries to satisfy the contextual similarity first and then tries to satisfy the spatial constraints using an auxiliary graph data structure. The running time of this algorithm is exponential in the number of objects in the query image. All of the examples in this paper are from the map domain. However, images from many other interesting applications fall into the category of symbolic images. These include CAD/CAM, engineering drawings, floor plans, and more. Hence, the methods that we describe here are applicable to them as well.

2 Pictorial Query Specification

To specify queries pictorially, the user creates an image containing the required symbols positioned so that the desired spatial constraints hold. The user must also specify the image similarity level required to satisfy the contextual and spatial constraints. Throughout this paper we use the following definitions and notation. A *symbol* is a group of connected pixels that together have some common semantic meaning. A *class* is a group of symbols that all have the same semantic meaning. $cl(s)$ denotes the class of symbol s. We say that symbol $s1$ *matches* symbol $s2$ if $cl(s1) = cl(s2)$.

We define the following four levels of contextual similarity between a query image QI and a database image DI:

1. The set of classes of all symbols in both images is the same.
2. For any symbol $s \in QI$ there exists a symbol $t \in DI$ such that $cl(s) = cl(t)$. (DI includes QI).
3. For any symbol $s \in DI$ there exists a symbol $t \in QI$ such that $cl(s) = cl(t)$ (QI includes DI).

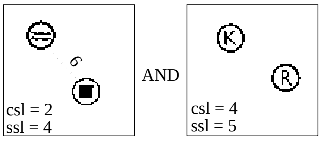

Fig. 1. Example query1

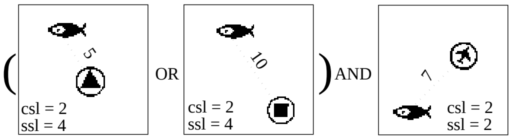

Fig. 2. Example query 2

4. There exists at least one symbol $s \in DI$ and one symbol $t \in QI$ such that $cl(s) = cl(t)$ $(DI \cap_{classes} QI \neq \emptyset)$.

In addition, we define the following five levels of spatial similarity:

1. The matching symbols of DI and QI (i.e., symbols that belong to the same class) are in exactly the same locations.
2. Matching symbols have the same inter-symbol spatial relationship, and the distance between them is bounded from below by L and bounded from above by the distance between the symbols in QI. $L = 0$ by default.
3. The distance between the matching symbols may vary, but the inter-symbol spatial relationship between them must be the same.
4. The inter-symbol spatial relationship between the matching symbols may vary, but the distance between them is bounded from below by L and bounded from above by the distance between the symbols in QI. $L = 0$ by default.
5. The distance and the inter-symbol spatial relationship between the matching symbols may vary (i.e., no spatial constraints).

The total similarity between QI and DI is defined by combining the two similarity factors. For example, $DI \equiv_{2,3} QI$ denotes that the contextual similarity and the spatial similarity of the two images is at levels 2 and 3, respectively. That is, for each symbol in QI there is a matching symbol from the same class in DI, the location of the symbols and the distance between them may vary, but the inter-symbol spatial relationship between them is the same. In general, if $DI \equiv_{csl,ssl} QI$ and if S' is the set of all the symbols of DI that match some symbol in QI, then the set of classes of the symbols of S' is a subset of the set of classes of the symbols of QI. Furthermore, for every pair of symbols s_1 and $s_2 \in S'$ the spatial constraints dictated by ssl and the positions of the matching symbols in QI hold.

More complex queries may be specified by combining query images with "AND" and "OR" operators. For example, to specify a query with spatial constraints between some symbols, but with no spatial constraints between other symbols, two separate query images with different values of ssl can be combined via the AND operator. For example, consider the query in Figure 1 which requests "all images with a hotel within 6 miles of a beach and with a cafe or a restaurant". No spatial constraints are specified for the restaurant and cafe symbols; however, the hotel must be within 6 miles of a beach. This method of splitting a query into several components can also be used to specify more than one acceptable spatial constraint. For example, Figure 2 requests "all images with a camping site within 5 miles of a fishing site OR with a hotel within 10 miles of a fishing site AND with an airfield northeast of and within 7 miles of the fishing site". Note that the dotted lines with the distance that appear in the query images in Figures 1 and 2 are only used to denote the distance between symbols in the figure; they are not actually part of the query image. The query image only contains symbols. The distance (and relative directions) between the symbols is specified implicitly in the query image QI by the actual distance (and relative direction) between the symbols in QI.

3 Pictorial Query Processing

```
checkSsl(logical image DI, QI, similarity level ssl)
if ssl = 5 ∨ |DI| = 1 then /* no need to check anything */
  return TRUE
/* compute distances and relative location between QI symbols*/
foreach qel1 ∈ QI
  foreach qel2 ∈ QI − {qel1}
    if (ssl = 2) ∨ (ssl = 4) then
      dists[cl(qel1), cl(qel2)] ← getDist(loc(qel1), loc(qel2))
    if (ssl = 2) ∨ (ssl = 3) then
      relDirs[cl(qel1), cl(qel2)] ← getReldir(loc(qel1), loc(qel2))
/* now check that these hold in the input image */
foreach del1 ∈ DI
  foreach del2 ∈ DI − {del1}
    if (ssl = 2) ∨ (ssl = 4) then
      if getDists(loc(del1), loc(del2)) > dists[cl(del1), cl(del2)] then
        return FALSE
    if (ssl = 2) ∨ (ssl = 3) then
      if getReldirs(loc(del1), loc(del2)) ≠ relDirs[cl(del1), cl(del2)] then
        return FALSE
return TRUE /* everything is OK */
```

Fig. 3. Algorithm *checkSsl* to determine whether the spatial constraints dictated by a query image QI and spatial similarity level ssl hold in a logical image DI.

Pictorial queries are processed by a function called *GetSimilarImages* that takes as input the query image (QI), the contextual similarity level (csl), and the spatial similarity level (ssl). It returns the set of databases images B so that each image DI of B satisfies the pictorial query (i.e., $DI \equiv_{csl,ssl} QI$ for all $DI \in B$). *GetSimilarImages* constructs a set of candidate images from the database in which the contextual constraints hold and then invokes function *checkSsl* for each candidate image to determine if the spatial constraints dictated by ssl hold in it. An image in which the spatial constraints do not hold is removed from the candidate-image set. Pictorial queries involving more than one query component are executed by performing a separate pictorial query for each component and then computing the intersection of the results for components joined by an AND operator, and the union of the results for those joined by an OR operator.

Algorithm *checkSsl* determines whether the spatial constraints dictated by a query image QI and spatial similarity level ssl hold in an image DI. Figure 3 summarizes this algorithm. The input to *checkSsl* is QI' and DI' which are sub-images of the original QI and DI that contain only those symbols that were matched to each other by *GetSimilarImages* when checking the contextual constraints. Thus, the set of classes of the symbols of DI' and QI' is identical although $|DI'|$ may be larger than $|QI'|$ since DI and hence DI' may have multiple instances of the same class. The algorithm first computes the distance and/or the relative directions between the symbols of QI'. It then computes them for DI' and checks whether the required spatial constraints between each symbol pair in DI' hold. See Figure 4 for example sub-images QI' and DI'.

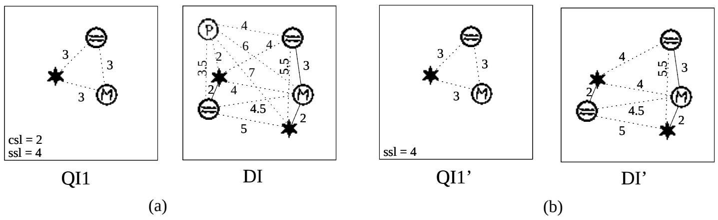

Fig. 4. (a) Query image $QI1$ and database image DI. (b) The corresponding sub-images $QI1'$ and DI' that contain only the matching symbols.

4 Multiple Symbol Instances in Database Images

Figure 5 illustrates the complexity that arises from allowing multiple instances of the same symbol in database images. Although there is one pair of symbols in DI' that satisfies the spatial constraints for each query-image pair (denoted by solid lines in the part of the figure corresponding to DI'), there is no triplet of symbols that does this. In other words, *checkSsl* cannot verify a triplet of conditions but it can catch the three pairs. Therefore, a pair-wise algorithm such as *checkSsl* cannot work here. Instead, a new and more complex version of *checkSsl* would be needed that checks every possible triplet combination. In general, for a query with $|QI'|$ symbols, we need to check for the occurrence of every possible combination of them. This new version of *checkSsl* would thus take $O(|QI'|^2+\binom{|DI'|}{|QI'|})$ time, which is exponential ($|QI'|^2$ to compute the distance between every pair of symbols in QI', and $\binom{|DI'|}{|QI'|}$ to check if these distances hold for any combination of size $|QI'|$ in DI').

One solution for this problem is based on the following definition and theorem. Note that the discussion here assumes that $ssl = 4$, i.e. the spatial constraints are only based on distance, however the results hold for cases involving relative directions as well.

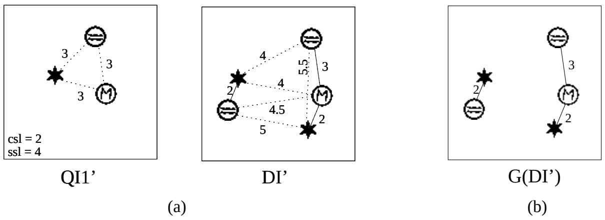

Fig. 5. (a) Query sub-image $QI1'$, and database sub-image DI' that does not satisfy the spatial constraint specified by ssl. (b) The corresponding query graph $G(DI')$ does not have a clique of size $|QI1'| = 3$.

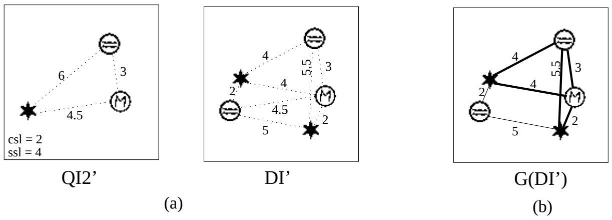

Fig. 6. (a) Query sub-image $QI2'$, and database sub-image DI' that does satisfy the spatial constraint. (b) The corresponding query graph $G(DI')$. It has two cliques of size $|QI2'| = 3$, indicated by thick lines.

Definition. A *clique* in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge of E. In other words, a clique is a complete subset of G. A clique of size k is a complete subgraph of G where $|V'| = k$.

Theorem 1. *Let $G(DI') = (V, E)$, where V = all symbols s in DI'. Let $(v_i, v_j) \in E$ if $dist(v_i, v_j)$ in DI' is less than or equal to $dist(cl(v_i), cl(v_j))$ in QI'. DI' satisfies the spatial constraints specified by QI' if and only if graph $G(DI')$ has a clique of size $|QI'|$.*

We first observe the use of this theorem by means of an example and then present a proof of it. Figure 5b shows graph $G(DI')$ that corresponds to the query $QI1'$ in Figure 5a. There is no clique of size $|QI1'| = 3$ in $G(DI')$, and thus DI' does not satisfy $QI1'$. However, for query sub-image $QI2'$, as seen in Figure 6, $G(DI')$ (in Figure 6b) has two cliques of size 3, and therefore DI' does satisfy $QI2'$. Recall that we are still assuming that there is only one instance of each symbol in the query image.

Proof. We first prove the "if" part (i.e., if $G(DI')$ has a clique of size $|QI'|$, then DI' satisfies the spatial constraints specified by QI'). If $G(DI') = (V, E)$ has a clique whose elements are $V' \subseteq V$ and if $|V'| = |QI'|$ then by definition of a clique each pair of elements of V' are connected by an edge of E. We denote this complete subgraph by $G'(DI')$. Each one of the vertices $v \in V'$ corresponds to exactly one of the symbols in QI'. The reason for this is that since we assume that QI (and thus QI') has only one instance of each symbol, and since the edges in $G(DI')$ correspond to constraints between symbols in QI', there are no edges in $G(DI')$ between multiple instances of the same symbol. Therefore, V' consists of $|QI'|$ unique symbols and there is a one-to-one correspondence between the vertices of $G'(DI')$ and the symbols of QI'. Furthermore, since $G'(DI')$ is a complete graph, all of the constraints dictated by QI' hold. Otherwise, the corresponding edge would not have been inserted into $G(DI')$. Thus, if $G(DI')$ has a clique of size $|QI'|$, then DI' satisfies the spatial constraints specified by QI'. We now prove the "only if" part (i.e., if DI' satisfies the spatial constraints specified by QI', then $G(DI')$ has a clique of size $|QI'|$). If DI' satisfies the spatial constraints specified by QI' then by the definition of $DI \equiv_{csl,ssl} QI$ in Section 2 and by the definition of DI' and QI' in Section 3 there exists a set of symbols $S' \in DI'$ where $|S'| = |QI'|$ in which for every pair of symbols s_1 and $s_2 \in S'$ the spatial constraints dictated by ssl for the matching symbols in QI' hold. According to the rule for constructing $G(DI')$, its set of vertices V contains one vertex for each symbol in DI'. In particular, it contains the set of vertices corresponding to the symbols in $S' \subseteq V$. Furthermore, an edge will be inserted between every two vertices corresponding to symbols in S'. Thus the subgraph of $G(DI')$ consisting of the vertices corresponding to the symbols of S' will be complete, and since $|S'| = |QI'|$, it is a clique of $G(DI')$ of size $|QI'|$.

Corollary 2. *$DI \equiv_{csl,ssl} QI$ if and only if DI satisfies QI in terms of csl and DI' satisfies QI' in terms of ssl.*

Proof. Immediate from the definitions of $\equiv$, DI' , and QI' and from Theorem1.

The *clique problem* is known to be NP-Complete [2], and thus we cannot expect to check for the spatial constraints using this method in polynomial time. However, the number of instances of query-image symbols in a database image is most likely not very large and is not a function of the size of the database. Instead, it is a function of the average number of symbols in an image which is a constant. Therefore, checking the spatial constraints by searching for cliques of size QI' is quite reasonable. The total cost of the modified version of *checkSsl* (called *McheckSsl*) that builds the graph and looks for cliques is $O(|DI'|^2+2^{|DI'|})$ where $|DI'|$ is the number of symbols in the database image that have matching symbols in the query image ($|DI'|^2$ for building graph $G(DI')$, and $2^{|DI'|}$ for searching for cliques of size $|QI'|$ in $G(DI')$). Note that this solution still assumes only one instance of each class in the query image.

5 Concluding Remarks

The method presented in this paper still assumes only one instance of each symbol in the query image. Handling multiple instances of symbols in query images is more complex since in this case each symbol in the database image may potentially be matched to more than one symbol in the query image. While one particular choice of matching may not yield a hit for the query, another choice may result in a positive result for the query. In addition, once a particular matching is chosen we must keep track of this binding in order to ensure that the other conditions hold with this particular binding. We are currently working on a solution for this problem.

Using our query specification method, the same query may be formulated in more than one way (by using different combinations of *csl* and AND/OR operators). One formulation may be more efficient to process than others. Furthermore, some combinations of AND and *csl* values may be impossible to satisfy in that the same image cannot satisfy both clauses of the AND. Therefore, query optimization methods for processing pictorial queries are required. This is a subject for future research.

References

1. S. K. Chang, Q. Y. Shi, and C. Y. Yan. Iconic indexing by 2-D strings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(3):413–428, May 1987.
2. M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., San Francisco, 1979.
3. V. Gudivada and V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Transactions on Information Systems*, 13(2):115–144, April 1995.
4. A. Gupta, T. Weymouth, and R. Jain. Semantic queries with pictures: the VIMSYS model. In G. Lohman, editor, *Proceedings of the 17th International Conference on Very Large Databases*, pages 69–79, Barcelona, Spain, September 1991.
5. R. C. Nelson and H. Samet. A consistent hierarchical representation for vector data. *Computer Graphics*, 20(4):197–206, August 1986. (also *Proceedings of the SIGGRAPH'86 Conference*, Dallas, August 1986).
6. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture, and shape. In *Proceeding of the SPIE, Storage and Retrieval of Image and Video Databases*, volume 1908, pages 173–187, San Jose, CA, February 1993.
7. D. Papadias and T. K. Sellis. A pictorial query-by-example language. *Journal of Visual Languages and Computing*, 6(1):53–72, March 1995.
8. A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *Proceeding of the SPIE, Storage and Retrieval of Image and Video Databases II*, volume 2185, pages 34–47, San Jose, CA, February 1994.
9. A. Soffer and H. Samet. Pictorial queries by image similarity. In *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, August 1996. To appear.
10. A. Soffer and H. Samet. Retrieval by content in symbolic-image databases. In *Proceedings of the SPIE, Storage and Retrieval of Still Image and Video Databases IV*, volume 2670, pages 144–155, San Jose, CA, February 1996.
11. M. Swain. Interactive indexing into image databases. In *Proceeding of the SPIE, Storage and Retrieval for Image and Video Databases*, volume 1908, pages 95–103, San Jose, CA, February 1993.
12. M. Ubell. The montage extensible dataBlade architecture. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, page 482, Minneapolis, MN, June 1994.