

# Evaluation of Spatial Similarity Methods for Image Retrieval

EURIPIDES PETRAKIS      COSTAS GEORGIADIS  
Department of Electronic and Computer Engineering  
Technical University of Crete  
Chania, Crete  
GREECE  
{petrakis, georgiad}@ced.tuc.gr

*Abstract:* Similarity retrieval by spatial content (i.e., using multiple objects and their interrelationships) in Image DataBases (IDBs) is still an open problem and has received considerable attention in the literature. In this work, we focus our attention on “queries by example” image and we study methods for answering such queries including the well accepted “*editing distance*” on Attributed Relational Graphs (ARGs) and the “*Hungarian Method*” for graph matching. We evaluate the time responses and the accuracy of these methods using a database of synthetic by realistic medical images. Our experiments indicate that the ARG editing distance, although the slowest, is the most accurate method.

*Key-Words:* Image DataBase, Image retrieval, Attributed Relational Graphs.

## 1. Introduction

The management of large volumes of digital images in many application fields (e.g., medicine [1], criminal investigation [2], trademark/copyright detection [3] etc.) has generated additional interest in methods and tools for real time archiving and retrieval of images by content. Consequently, content-based image retrieval has become the object of intensive research activities over the past few years [4, 5].

Several approaches to the problem of content-based image management have been proposed and some have been implemented on research prototypes and commercial systems [6, 7, 8, 9]. There are two general goals common to all IDB systems:

**Effectiveness:** An IDB system must retrieve similar images with as few errors as possible.

**Efficiency:** It must be *fast* to meet the speed requirements of an IDB environment.

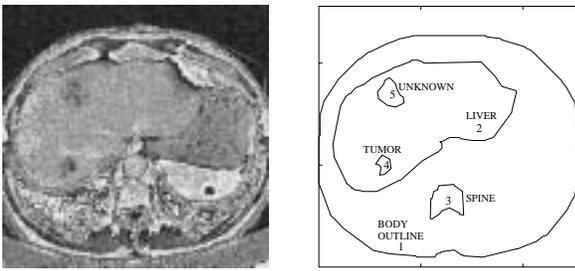
Focusing mainly on color, texture and shape, the work referred to above does not show how to handle multiple objects or regions in example queries, nor their inter-relationships. In this case, for two images to be similar, not only the shape, color and texture properties of individual image regions must be similar but, they must have the same arrangement (i.e., spatial relationships) in the two images. 2D strings [10] and “Attributed Relational Graphs” (ARGs) [1] are examples of methods focusing on spatial relationships. 2D string matching give binary (i.e., “yes/no”) answers and may yield “false alarms” (non-qualifying images) and “false dismissals” (qualifying but not retrieved images) [11].

In this work, we focus our attention on ARGs. We experimented with medical tomographic images but ARGs can handle any image type. Specifically, we deal with the following problem:

- We have a collection of  $N$  medical images showing multiple objects. Each image has been segmented (automatically or manually) to more than one objects or regions. Figure 1 shows an example of an original grey-level image (left) and its corresponding segmented form (right).
- The queries are “by example”: The user specifies a desirable image (e.g., “*find the examinations which are similar to Smith’s examination*”).
- The system has to return all images showing similar objects with the query in similar spatial relationships. There may exist extra objects in a retrieved image but not in the query.

The contributions of this work are summarized in the following:

- We study methods capable of answering spatial queries such as the well accepted “*editing distance*” on ARGs and the so called “*Hungarian Method*” for graph matching.



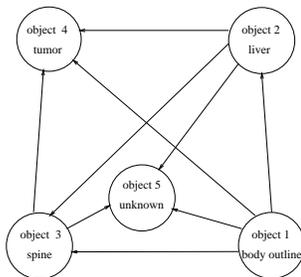
**Figure 1. Example of an original grey-level image (left) and its segmented form (right).**

- We evaluate the effectiveness (i.e., accuracy) of all these methods in terms of *precision* and *recall*. Our evaluation is based on human relevance judgements following a well established methodology from the information retrieval field.
- We evaluate the efficiency (i.e., time responses) of all these methods.

The rest of this paper is organized as follows: A short review of the underlying theory is presented in Section 2. Our evaluation method is discussed in Section 3. Experimental results are presented in Section 4 followed by conclusions in Section 5.

## 2. Attributed Relational Graphs (ARGs)

Figure 2 illustrates the ARG computed to the image of Figure 1. Nodes correspond to regions and arcs correspond to relationships between regions. The arcs are directed from the outer to the contained region but their direction also depends on object labels: Arcs are always directed (a) from body outline to the remaining objects, (b) from liver to spine and (c) from the most common objects (i.e., body outline, liver and spine in our application) to the remaining objects.



**Figure 2. Example ARG.**

Both nodes and arcs are labeled by attributes corresponding to properties (features) of objects and to properties of relationships respectively. Node and edge

features form vectors. A set of features that has been used successfully [11, 1] is the following:

**Individual regions:** *size*, computed as the size of the area it occupies, *perimeter* computed as the length of its bounding contour, *roundness*, computed as the ratio of the smallest to the largest second moment and *orientation*, defined to be the angle between the horizontal direction and the axis of elongation. This is the axis of least second moment.

**Spatial relationships:** The following features are used to describe the spatial relationships between regions: *relative distance*, computed as the minimum distance between their surrounding contours, *relative orientation* defined as the angle with the horizontal direction of the line connecting the centers of mass of their regions and *relative position* taking values  $(-1, 0, 1)$  corresponding to regions which are the one inside the other (-1), outside each other (0) or, the second inside the first one (1) respectively.

To avoid discontinuities in the measurement of angles (i.e., orientations of 1 degree should have measurements similar to those of orientations of 359 degrees), angles are represented by both *sin* and *cos*. All attributes are normalized in the range  $[-1, 1]$ : Angle attributes (i.e., *sin*, *cos* take values in the range  $[-1, 1]$ ; distances and areas are normalized in the range  $[0..1]$  by dividing them respectively by the perimeter and the area of the largest region (i.e., body outline in our case). Notice that, an ARG representation can handle any set of features as labels.

### 2.1. ARG Editing Distance

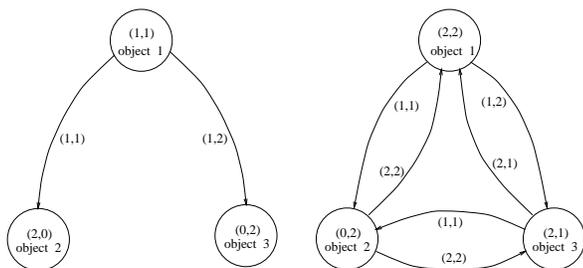
Matching a query and a stored ARG is treated as an “*error-correcting subgraph isomorphism*” problem [12]: Given two graphs  $G$  (query) and  $G'$  (stored or reference ARG) there is always a sequence of “edit” operations that transform  $G$  to a subgraph of  $G'$  (i.e., there may exist extra nodes in  $G'$ ). These edit operations take the form of node or edge insertions, deletions and substitutions.

The calculation of the distance between two ARGs involves not only finding a sequence of edit operations, but also finding the one that yields the minimum total cost. This can be formulated as a tree search problem which is solved by an  $A^*$  algorithm [13]: Each tree node corresponds to a matching of a pair of subgraphs from the two input ARGs. A transition from a state to another corresponds to the embedding (matching) of a pair of unmatched nodes (one from each ARG) and of

their associated edges, into the already matched sub-graphs. The matching operations for each embedding are recorded and their costs are accumulated. The tree expands until a complete match is found. A complete match is one that has consumed all nodes of  $G$  (but not necessarily all nodes of  $G'$ ).

The method for ARG matching referred to above finds the optimal solution but it has exponential time (and space) complexity in the worst case (i.e., the matching tree can grow exponentially for large graphs). Approximate methods with lower time complexity do exist [14, 15] but they are not guaranteed to find the optimal solution.

Figure 3 shows an example of a query ARG (left) and of a reference (model) ARG (right). All nodes and all edges are labeled by attribute vectors.

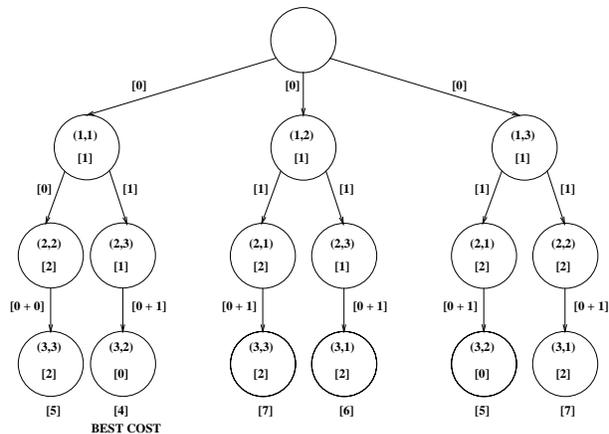


**Figure 3. Example query ARG (left) and reference ARG (right).**

The algorithm creates the matching tree of Figure 4. Each node of Figure 4 is labeled by a pair of matched nodes (in parenthesis) and by the cost of matching these nodes (in square brackets). Transitions are labeled by the costs of matching the relationships of the added nodes with all the nodes currently on the path. Notice that only substitution costs are used (i.e., no nodes or edges are inserted or deleted). The cost of substituting a node or edge is computed as the distance (i.e., Chebyshev distance in this example) between their corresponding feature vectors. Node and edge costs along the same path are summated. In this example, query node 1 is associated with model node 1, query node 2 is associated with model node 3 and query node 3 is associated with model node 2. The cost of this matching (best cost) is 4.

## 2.2. Hungarian Method

Matching between two ARGs can also be formulated as an *assignment* problem, that is a problem of finding the best association between the nodes (objects) of a query  $Q$  and the nodes (objects) of a reference



**Figure 4. Matching tree.**

image  $I$  (the relationships are ignored). To compute this association, we need the costs (weights)  $C(i, j)$  of associating each object  $i$  in  $Q$  with each object  $j$  in  $I$ ,  $i, j \leq n$ , where  $n$  is the number of objects in the two images.  $C(i, j)$  is computed as the distance between their corresponding feature vectors.

Let  $F()$  be a mapping from objects in  $Q$  to objects in  $I$ . The cost of this mapping is defined as:

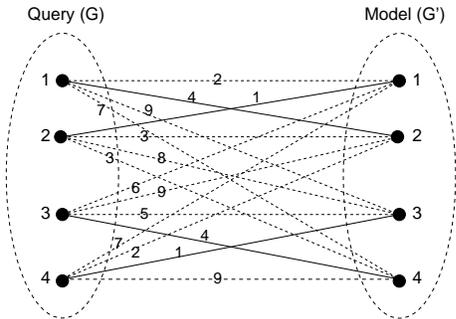
$$D_F(Q, I) = \sum_{i=1}^n C(i, F(i)). \quad (1)$$

The distance between images  $Q$  and  $I$  is defined as the minimum distance computed over all possible mappings  $F()$  and corresponds to the best association of objects in  $Q$  with objects in  $I$ :

$$D(Q, I) = \min_F \{D_F(Q, I)\}. \quad (2)$$

Figure 5 illustrates a bipartite graph which is constructed from a query  $Q$  and a reference image  $I$  by associating each object in  $Q$  with all objects in  $I$ . The labels on the arcs connecting objects in  $Q$  with objects in  $I$  correspond to the costs  $C(i, j)$ . The best mapping  $F()$  is computed using the *Hungarian* method in  $O(N^4)$  time [16]. Solid lines in Figure 5 correspond to the best mapping. The cost of this mapping is 10. This is the cost of associating object 1 in  $Q$  with object 2 in  $I$ , plus the cost of associating object 2 with object 1, plus the cost of associating object 3 with object 4 and, finally, object 4 with object 3.

The above formulation assumes that the two images have the same number of objects. However, if  $Q$  contains fewer objects than  $I$ , we add any missing objects in  $Q$ , assuming that the costs  $C(i, j)$  incident upon them are all  $\infty$ . If  $Q$  contains more objects than  $I$ , then,  $D(Q, I)$  is  $\infty$  by definition (i.e., extra objects are not allowed in  $Q$ ).



**Figure 5. Image matching as an assignment problem.**

### 2.3. Hungarian Plus

The Hungarian method shows how to find the best association between the objects in a query and a database image but without handling their interrelationships. Gudivada and Raghavan [17] assume that such an association is given and they compute the cost of this association by taking the differences of the relative orientations between all pairs of matched objects in the two images. In particular, the cost  $D(Q, I)$  between a query  $Q$  and a reference image  $I$  is computed as

$$D(Q, I) = 1 - \sum_{i,j \leq [1,n]} \frac{1}{n(n-1)/2} \frac{1 + \cos(\theta_{i,j} - \theta_{F(i),F(j)})}{2}, \quad (3)$$

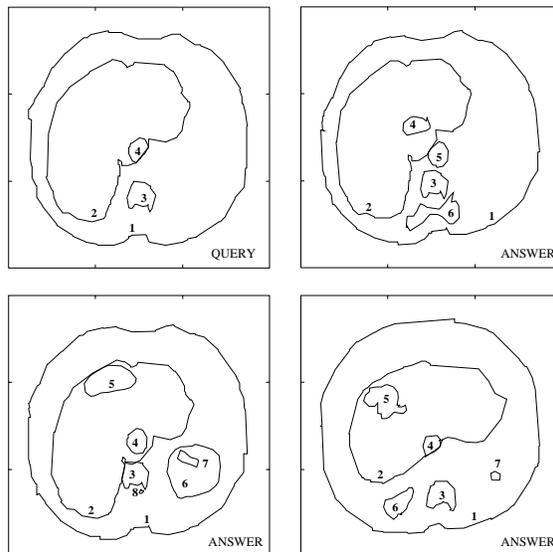
where  $\theta_{i,j}$  is the relative orientation between objects  $i, j$  in  $Q$  and  $\theta_{F(i),F(j)}$  is the relative orientation between their associated objects  $F(i), F(j)$  in  $I$ . This formulae takes the summation of all such differences and then takes their average by normalizing with the number of relationships which is  $n(n-1)/2$ .

Notice that [17] doesn't show how to compute a mapping  $F()$ . We propose that the Hungarian method is used to compute this mapping. Then, the cost of matching two images is computed according to Equation 3. This method is called henceforth "*Hungarian Plus*".

### 3. Evaluation Criteria

We used human relevance judgments to compute the effectiveness (i.e., accuracy) of each method. These relevance judgments have been suggested by a radiologist: Two images are considered similar if they contain similar objects in similar spatial relationships. If the objects are significantly different, the images are considered dissimilar. However, for two objects to be considered as similar, exact shape similarity is not required. Figure 6 illustrates a typical query (top left

image) specifying 4 objects and 3 qualifying images. Objects 1, 2, 3 and 4 in the query image are matched with objects with the same indices in the retrieved images. In this example, for simplicity, we show all associated objects with the same indices. Notice that a retrieved image (but not the query) may contain extra objects.



**Figure 6. Example of a query image (top left) and of three qualifying images.**

For each candidate method we computed:

**Precision** defined as the percentage of similar images retrieved with respect to the total number of retrieved images.

**Recall** defined as the percentage of similar images retrieved with respect to the total number of similar images in the database. Because we don't have the resources to compare every query with all database images, for each query, we merged the answers we obtained by all methods and we considered this as the database which is manually inspected for similar images. This is a valid sampling method known as "*pooling method*" [18]. This method does not allow for absolute judgments such as "method  $A$  misses 10% of the total number of similar images in the database". It provides, however, a fair basis for comparisons between methods allowing judgments such as "method  $A$  returns 5% fewer correct answers than method  $B$ ".

## 4. Experiments

We implemented all methods in *C++* and we run all our experiments on a dedicated SUN Ultra 1 running Solaris<sup>TM</sup> 167MHz. As a testbed, we used the dataset of 13,500 synthetic segmented images<sup>1</sup> of [1]. To evaluate our methods we created 20 characteristic queries. All images and the queries contain between 4 and 8 objects.

The experiments are designed to demonstrate the:

**Effectiveness** (i.e., accuracy) of each candidate method.

We selected the best method based on measurements of precision and recall.

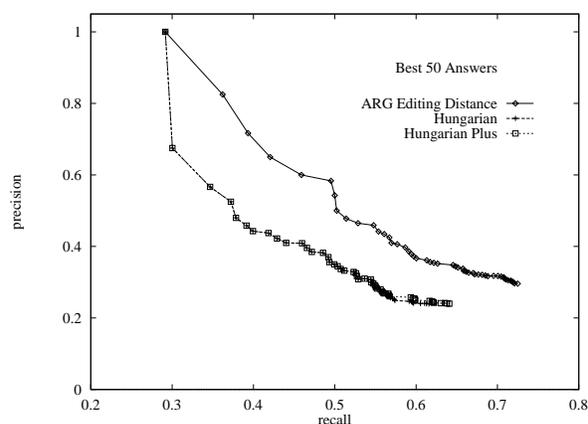
**Efficiency** (i.e., speed) of each candidate method. We computed the elapsed (wall-clock) times using the `time` system call of UNIX and we selected the faster method. These are times for database search; times for image display are not taken into account since they are common to all methods.

### 4.1. Effectiveness

The purpose of this set of experiments is to identify the most accurate method. We present a *precision-recall plot* for each candidate method. The horizontal axis in such a plot corresponds to the measured recall while, the vertical axis corresponds to precision. Each query retrieves the best 50 answers (best matches) and each point in our plots is the average over 20 queries. Each method in such a plot is represented by a curve. The top-left point of a precision/recall curve corresponds to the precision/recall values for the best answer (best match) while, the bottom right point corresponds to the precision/recall values for the entire answer set.

ARG editing distance is the most accurate method achieving up to 10% better precision and recall than any other method. Hungarian and Hungarian Plus are roughly equivalent with Hungarian Plus achieving slightly better recall. The Hungarian method always retrieves images with similar objects which, in most cases, happen to have the desired spatial relationships. Hungarian Plus takes the output of the Hungarian method in its input and, simply, rearranges the order of the above answers by inspecting their spatial relationships using Equation 3. However, Hungarian Plus introduces only a few new images within the first 50 answers.

<sup>1</sup>The test data are available from `ftp://ftp.ced.-tuc.gr/pub/petrakis`.



**Figure 7. Precision-Recall diagram corresponding to (a) ARG Editing Distance (b) Hungarian and (c) Hungarian Plus methods.**

### 4.2. Efficiency

The purpose of this set of experiments is to identify the faster method. Each method is used to search the IDB sequentially (i.e., the query is matched with all stored images). Table 1 shows the average (i.e., over 20 queries) retrieval response times obtained for each method.

ARG Editing Distance	Hungarian	Hungarian Plus
23.2 secs	13.6 secs	16.5 secs

**Table 1. Average retrieval response times (in seconds).**

ARG matching is obviously the slowest method (i.e., ARG matching has, in general, exponential time complexity). Hungarian, is the fastest method because of its polynomial time complexity. Hungarian Plus, on the other hand, is slower than the plain Hungarian method because it requires additional processing for the computation of the spatial relationships.

## 5. Conclusions

We evaluated three methods for answering queries by example image on an image database namely, the ARG editing distance, Hungarian and Hungarian Plus. Our experiments indicate that the ARG editing distance, although the slowest, has certain advantages over all other methods:

- It is the most accurate demonstrating higher *precision* and *recall* than any other method.

- It is extendible: It can accommodate any type and number of properties of regions or relationships that a domain expert may appreciate.

Hungarian method, is in turn, preferred over Hungarian Plus: It is almost as accurate as Hungarian Plus but, much faster. Future work includes the experimentation with more methods (e.g., 2D strings) and the investigation of indexing methods for speeding-up the retrieval response times of these methods.

## Acknowledgements

We are grateful to Charalambos Genzis for his help in the experiments.

## References

- [1] Euripides G.M. Petrakis and Christos Faloutsos. Similarity Searching in Medical Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):435–447, May/June 1997.
- [2] Nalini K. Ratha, Kalle Karu, Shaoyun Chen, and Anil K. Jain. A Real Time Matching System for Large Fingerprint Databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):799–813, August 1996.
- [3] Anil K. Jain and Aditya Vailaya. Shape-Based Retrieval: A Case Study With Trademark Image Databases. *Pattern Recognition*, 31(9):1369–1390, 1998.
- [4] William I. Grosky. Management Multimedia Information in Database Systems. *Communications of the ACM*, 40(12):73–80, 1997.
- [5] Alberto Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc., 1999.
- [6] Amarnath Gupta and Ramesh Jain. Visual Information Retrieval. *Communications of the ACM*, 40(5):71–79, May 1997. (<http://www.virage.com>).
- [7] M. Flickner et. al. Query By Image and Video Content: The QBIC System. *Computer*, 28(9):23–32, September 1995. (<http://www.qbic.almaden.ibm.com>).
- [8] A. Pentland, R. W. Picard, and A. Sclaroff. Photobook: Content Based Manipulation of Image Databases. *International Journal of Computer Vision*, 18(3):233–254, 1996. (<http://vismod.www.media.mit.edu/vismod-/demos/photobook/index.html>).
- [9] John R. Smith and Shih-Fu Chang. VisualSEEK: A Fully Automated Content Based Image Query System. In *Proceedings of ACM Multimedia Conference*, Boston Ma., November 1996. (<http://disney.ctr.columbia.edu/VisualSEEK>).
- [10] S.-K. Chang, E. Jungert, and G. Tortora, editors. *Intelligent Image DataBase Systems*. World Scientific, 1996.
- [11] Euripides G.M. Petrakis and Stelios C. Orphanoudakis. Methodology for the Representation, Indexing and Retrieval of Images by Content. *Image and Vision Computing*, 11(8):504–521, October 1993.
- [12] B. T. Messmer. *Efficient Graph Matching Algorithms*. PhD thesis, University of Bern, Switzerland, 1995. (<http://iamwww.unibe.ch/~fkiwww/projects/GraphMatch.html>).
- [13] N. J. Nilsson. *Principles of Artificial Intelligence*. Tioga, Palo Alto, 1980.
- [14] William J. Christmas, Josef Kittler, and Maria Petrou. Structural Matching in Computer Vision Using Probabilistic Relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, August 1995.
- [15] H. A. Almohamad and S. O. Duffuaa. A Linear Programming Approach for the Weighted Graph Matching Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):522–525, May 1993.
- [16] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter 11, pages 247–255. Engelwood Cliffs: Prentice Hall, 1982.
- [17] Venkat N. Gudivada and Vijay V. Raghavan. Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity. *ACM Transactions on Information Systems*, 13(2):115–144, 1995.
- [18] D.K. Harman, editor. *The Second Text Retrieval Conference*. National Institute of Standards and Technology Gaithersburg, MD, 1994.