Content-Based Retrieval of Medical Images

EURIPIDES G.M. PETRAKIS
Department of Electronic and Computer Engineering
Technical University of Crete
Chania, Crete
GREECE
e-mail: petrakis@ced.tuc.gr, URL: http://www.ced.tuc.gr/~petrakis

Abstract: We consider the requirements for the design and implementation of Image DataBase (IDB) systems which support the retrieval of medical images by content. Attention is focused on a methodology for the efficient representation and retrieval of medical images based on spatial information. The content of medical images is represented by Attributed Relational Graphs (ARGs) holding features of objects or regions and relationships between such objects or regions. The method can answer queries by example, such as *"find all image examinations similar to Smith's examination"*. The performance of the method has been evaluated using a dataset of 13,500 simulated, but realistic computed tomography and magnetic resonance images.

Key-Words: image database, medical image, attributed relational graph, query by example.

1 Introduction

The medical imaging field has grown substantially in recent years and has generated additional interest in methods and tools for the management, analysis, and communication of medical images. Many diagnostic imaging modalities, such as X-ray computed tomography (CT), magnetic resonance imaging (MRI), digital radiography, and ultrasound are currently available and are routinely used to support clinical decision making. It is important to extend the capabilities of such application fields by developing image database systems supporting the automated archiving and retrieval of medical images by content. In addition, medical IDB systems are emerging as an important component of Picture Archiving and Communications Systems (PACS) in order to support administrative, clinical, teaching and research activities.

Several approaches to the problem of content-based image management have been proposed and some have been implemented on research prototypes and commercial systems: In the Virage system [1], image content is given primarily in terms of properties of color and texture. The QBIC system [2] of IBM utilizes a retrieval method for queries on color, texture and shape. The Photobook [3], the system developed at the MIT Media Lab, supports queries by image content in conjunction with text queries.

Regarding medical IDB systems, Chu et.al. [4] describes a system for retrieving medical images by spatial and temporal content. A knowledge-based approach for representing the content of medical images in an IDB is proposed as well. Commaniciou et.al. [5] proposes a system whose purpose is support of decision making in clinical pathology. Decision making is guided by the results of image segmentation and retrieval. I^{2}Cnet [6], the system developed at FORTH, is a network of content-based similarity search engines and supports retrievals on medical image databases based on attributes and text, in conjunction with geometric and texture properties of selected regions.

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. ARGs [7] and 2D strings [8] are examples of work focusing mainly on spatial relationships.

Three are the main contributions of this work towards developing a medical IDB system:

- Identifies the requirements for the design and development of medical IDB systems for CT and MRI images.
- Proposes a methodology for the efficient representation and retrieval of medical images by content based on Attributed Relational Graphs (ARGs).
- Examines the effectiveness of ARGs for the retrieval of CT and MRI medical images by content.

It is proposed that an IDB system consists of a user friendly interface and graphical tools facilitating the interaction between the user and the various system components. For example, the user is allowed to interact with the IDB and correct the results of image segmentation. The purpose of image segmentation is to identify and extract regions (or objects) of interest from all images, a task if crucial importance for the correct characterization of image content in any IDB system. The user is also allowed to specify the class to which an image belongs. Then, the system may resume responsibility for the efficient representation, storage and retrieval of images using properties of individual regions and spatial relationships between such regions.

The rest of this paper is organized as follows: A short presentation of the underlying theory on ARGs is presented in Section 2. An approach for the design of an IDB environment for medical images is discussed in Section 3. An implementation of the above environment along with experimental results are discussed in Section 4, followed by conclusions and issues for future research in Section 5.

2 Attributed Relational Graphs (ARGs)

Given a collection of images we must derive appropriate representations of their content and organize the images together with their representations in the database so that we can search efficiently for images similar to an example image. Image descriptions are given in terms of object properties and in terms of relationships between objects. The textbook approach to capture this information is the "Attributed Relational Graphs (ARGs)" or (simply) "graphs" [9].

Definition 1 *A graph G is a tuple $(V, E, A_V, A_E, \mu, \nu)$ where,*

- $V = \{v_1, v_2, \dots v_n\}$ *is a finite set of n nodes and* $E = \{e_1, r_2, \dots e_m\} \subseteq V \times V$ *is a finite set of m edges. An edge e_k, $1 \leq k \leq m$, is an ordered pair of nodes (v_i, v_j), $i \neq j$ and $1 \leq i, j \leq n$, denoting an arc from v_i to v_j.*
- A_V *and* A_E *are alphabets of node and edge attributes respectively.*
- μ *and* ν *are functions (or sets of functions) for generating node and edge attributes respectively.*

The above definition assumes directed edges: An edge (v_i, v_j) is directed from node v_i to node v_j. An undirected edge between nodes v_i and v_j is equivalent to two directed edges, one from v_i to v_j and one from v_j to v_i.

In an ARG representation of an image, objects are represented by graph nodes and relationships between objects are represented by arcs between such nodes. Both nodes and arcs are labeled by attributes corresponding to properties (features) of objects and relationships respectively. Figure 1

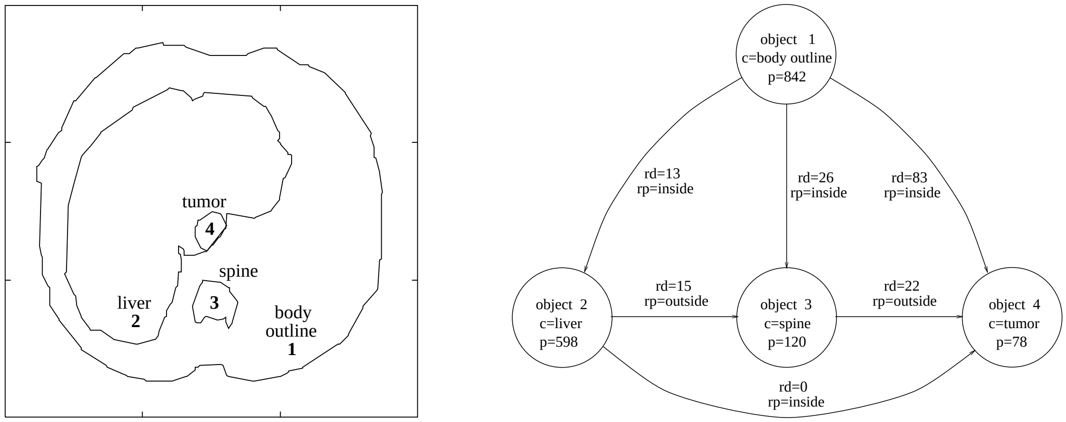

Figure 1: *Example of a medical image (left) and its corresponding ARG (right).*

shows an example image (i.e., a sketch of a medical image illustrating body outline, liver, spine and a tumor object) and its corresponding ARG $G = (V, E, A_V, A_E, \mu, \nu)$ where, $V = \{1, 2, 3, 4\}$ ($n = 4$), $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$, $A_V = \{c, p\}$, $A_E = \{rd, rp\}$. Finally, $\mu = \{name(v_i), perimeter(v_i)\}$ consists of functions that compute the name and the perimeter of each object v_i, and $\nu = \{relative\_distance(v_i, v_j),\ relative\_position(v_i, v_j)\}$ consists of functions that compute the minimum distance and the relative position between all pairs of objects v_i and v_j.

Notice that, an ARG representation can handle any set of features as labels. In the case of medical CT and MRI images, the set of labels used in this work is given in Section 4. In general, μ and ν define grey-level and texture values, moments or Fourier coefficients etc., as node labels, relative size, relative orientation, amount of overlapping or adjacency etc., as edge labels.

A query and a stored (model) image are considered similar if they contain similar objects with similar relationships. There may exist extra objects in a stored image but not in the query. The problem of retrieving images which are similar to a given example image is transformed into a problem of searching a database of stored ARGs: Given a query, its ARG G is computed and compared with all stored ARGs G'. In this work, matching a query and a stored graph is treated as a *"error-correcting subgraph isomorphism"* problem [9]:

Definition 2 *An "error-correcting subgraph isomorphism" from G to G' is defined by a sequence of edit operations $S(G')$ that transform G' to G.*

Given two graphs G and G' there is always a sequence $S(G') = \delta_k(\delta_{k-1}(\ldots\delta_2(\delta_1)\ldots))$ of k edit operations that transform G' to G (i.e., to a subgraph of G' which is isomorphic to G). Notice that, k need not be equal in all sequences $S(G')$. These edit operations take the form of node or edge insertions, deletions and substitutions. There are infinite sequences of such edit operations and one would like to choose the *best* one. This can be achieved by assigning costs to edit operations, combining the costs of a sequence $S(G')$ in a meaningful way and by taking the sequence yielding the minimum cost.

Definition 3 *The "distance" $D(G, G')$ between two graphs G and G' is defined as the minimum cost taken over all sequences of edit operations ("error corrections") that transform G' to G:*

- $D(G, G') = \min_{S(G')}\{\Phi(S(G'))\} = \min_{S(G')}\{\Phi(\phi(\delta_k), \phi(\delta_{k-1}), \ldots\phi(\delta_1))\}$ where,
- Φ is a function that combines the costs of all the k edit operations δ in $S(G')$, and
- ϕ is a function that computes the cost of an edit operation δ_i, $1 \leq i \leq k$.

The definition of cost functions Φ and ϕ depends highly on the application, the edit operations allowed and on the labels used: Φ can be defined either as a summation of ϕ costs:

$$\Phi(S(G')) = \sum_{i=1}^{k} \phi(\delta_i) \tag{1}$$

or as a max operation on the ϕ costs:

$$\Phi(S(G')) = \max_{i=1}^{k} \{\phi(\delta_i)\} \, . \tag{2}$$

In this work, we adopted the later definition ($\max$ tends to give low values of distance when *all* cost functions take low values). If μ and ν generate attribute vectors as node and edge labels (as it happens in our application) then, ϕ can be defined as a vector distance which is computed using an L_p metric. If $\mu(v) = (z_1, z_2, \ldots z_w)$ and $\mu'(v') = (z'_1, z'_2, \ldots z'_w)$ are two such feature vectors (for two nodes, one from G and one from G') then

$$\phi(\delta) = \phi(\mu(v) \rightarrow \mu'(v')) = \left[\frac{1}{w}\sum_{i=1}^{w} |z_i - z'_i|^p\right]^{1/p} , \tag{3}$$

where p is the order of the metric. For $p = 1$ and $p = 2$ we obtain the Manhattan (city-block) and the Euclidean distance respectively. If $p \rightarrow \infty$, we obtain the Chebeychev distance:

$$\phi(\delta) = \phi(\mu(v) \rightarrow \mu'(v')) = \max_{1 \leq i \leq w} \{|z_i - z'_i|\} \, . \tag{4}$$

We adopted the Chebeychev distance for ϕ. For edges, $\delta(\nu(e) \rightarrow \nu'(e')$ is defined similarly. In this work, we are mainly interested in substitution costs. Extra nodes or edges in G' (i.e., in a stored graph) are ignored (i.e., their cost is 0) while, extra nodes or edges in Q (i.e., in a query) are not allowed (i.e., their cost is ∞).

The computation of the distance between two ARGs involves not only finding a sequence of error transformations, but also finding the one that yields the minimum total cost. This can be formulated as a tree search problem which can be solved by an $A^\star$ algorithm [9]. In the following, the ARG at the left of Figure 2 (query ARG) is matched with the ARG at the right (model ARG). All nodes and all edges are labeled by their attribute vectors consisting of two attributes taking values in the range [0,2].

The algorithm creates the state-space tree of Figure 3. Each state (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 of a pair of unmatched nodes (one from each ARG) into the already matched subgraphs. Each state in Figure 3 is labeled by a pair of 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.

The state-space tree expands until a complete match is found. A complete match is one that has consumed all query nodes (but not necessarily all model nodes). The minimum total cost found at any time can be used as an upper bound to prune the expansion of feature non-promising paths yielding partial cost greater than the minimum total cost found so far.

The edit operations for each embedding are recorded and their costs are accumulated in a meaningful way. Node or edge matching costs are computed by taking the Chebeychev distance of their attribute vectors. Node and edge costs along the same path are summated. In the above 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 is 4 (best cost). Alternatively, we can take

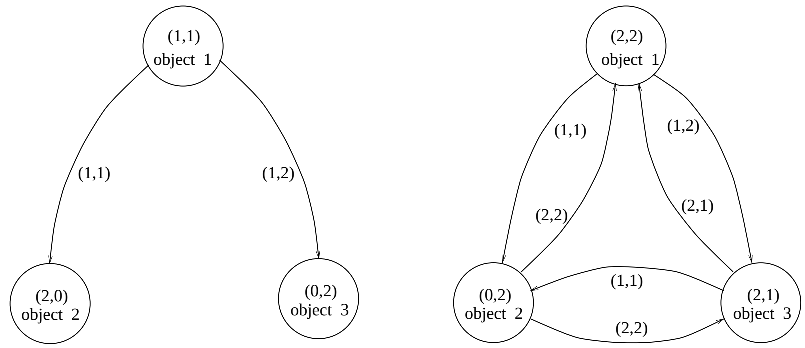

Figure 2: *Example of a query ARG (left) and a model ARG (right).*

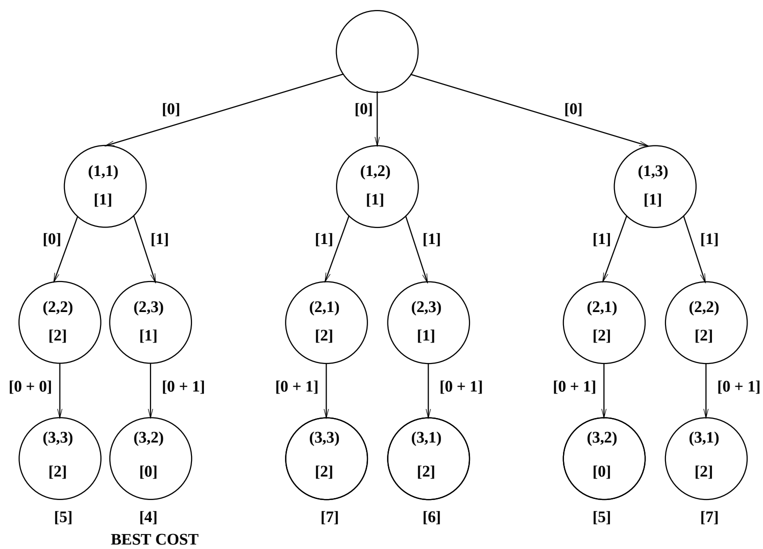

Figure 3: *State-space tree of ARG matching.*

the maximum of all the node and edge costs along a path. In this case, the best cost is 1 (all other paths have cost 2).

The method for ARG matching referred to above finds the optimal solution but it has exponential time and space complexity in the worst case. Approximate methods with lower time and space complexity do exist but they are not guaranteed to find the optimal solution (e.g., [10]).

3 Medical IDB Environment

Before any required image descriptions are extracted and used, images must first be segmented into disjoint regions or objects. Figure 4 shows an example of an original grey-level image (left) and its corresponding final segmented polygonal form (right).

The segmentation of CT and MRI images is in general very difficult and it is currently the subject of independent research activities (e.g., [4, 5]). However, it should be pointed out that the requirement for accurate and robust image segmentation is more relaxed for retrieving images by content than it is for image analysis and image understanding. Thus, we are opted for a conventional image segmentation technique resulting in polygonal approximations of object contours. We also propose that segmentations are carried out under the supervision of a domain expert (i.e., a clinician). First, images

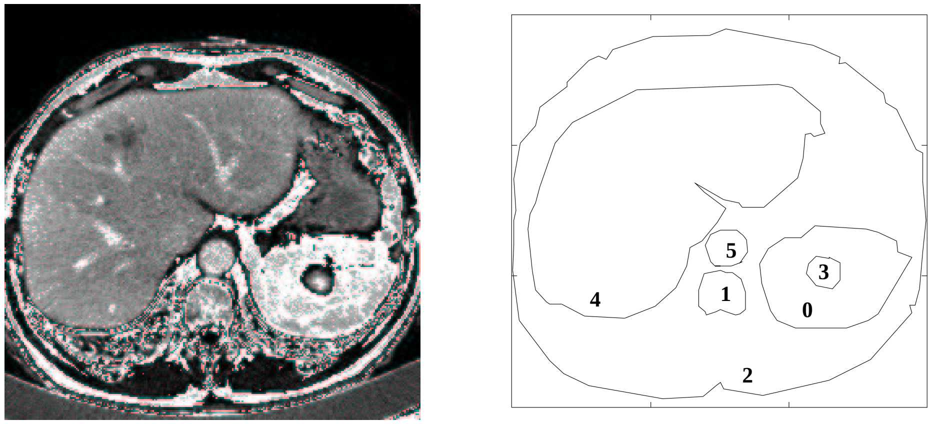

Figure 4: *Example of an original grey-level image (left) and its segmented form (right).*

are segmented using low-pass filtering followed by edge detection. A polygonal approximation to the derived segment edges is then obtained. The desired segmentation results are then obtained by editing (i.e., the expert may delete insignificant segments or correct the shape of others).

The role of the derived segmented forms is then twofold: First, they are used to compute a variety of image features specific to a particular image representation; second, they are stored in the database together with the original grey-level images and may be used for browsing the retrieved images.

Images may be classified into one or more predefined anatomical classes. Such classes may be defined based on parts of the body (e.g., head, neck, etc.) and/or the image plane position and orientation (e.g., axial, sagittal coronal slices of the head etc.). Furthermore, an image or a part of an image (i.e., an object, a segment or a region) may be classified into one or more predefined classes corresponding to normal or abnormal anatomical structures (e.g., ventricle, tumor, hematoma, etc.). Classes may be organized into anatomical and diagnostic hierarchies. Knowledge in the form of procedures (e.g., image processing and retrieval procedures corresponding to a specific class), rules, and parameters may be assigned to each class and may be inherited by the lower level classes. Figure 5 shows an anatomical hierarchy [11]. The higher level of abstraction is on the left and the lower level is on the right. Higher level classes inherit the instances of all their descendant (lower level) classes.

We assume that image classifications are carried-out interactively by a domain expert: Once an image is given, it can be classified into an appropriate anatomical class by selecting its name or icon from the class hierarchy. In this work, we illustrate the effectiveness of our proposed method of image content representation and retrieval focusing on MRI images of the abdomen. Figure 4 illustrates a representative image of this class. However, our proposed method can be applied to any image class of this anatomy hierarchy.

The retrieval capabilities of an IDB must be embedded in its query language. Query formulation needs to be iterative and flexible, enabling gradual resolution of user uncertainty. All images (and/or information related to images) satisfying the selection criteria should be retrieved and displayed for viewing. Furthermore, a query response can be refined by *"browsing"*.

The highest complexity of image queries is encountered in queries by example: A sample image or sketch is provided and the system must analyze it, extract an appropriate representation of its content, match this representation against representations of images stored in the database and, finally, retrieve all images with similar representations. Other types of image queries include, queries by conditional statements involving various image attributes (i.e., values of attributes and/or ranges of such values), queries by identifier (i.e., a unique key is specified), region queries (i.e., an image region is specified and all regions that intersect it are returned), text queries (i.e., a phrase or caption related to diagnosis

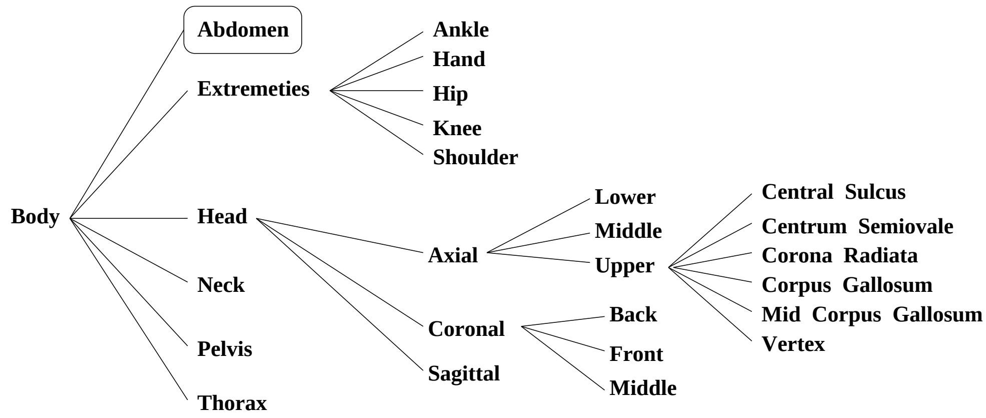

Figure 5: *Anatomy hierarchy defined.*

is specified) etc.

4 Implementation

In the following, we present an implementation of a prototype medical IDB retrieval system. Attention is focused on the representation of medical images by Attributed Relational Graphs (ARGs) and on retrievals by image content using the error-correcting distance measure on ARGs defined in Section 2.

All images are segmented into closed contours corresponding to dominant image objects or regions, and all image components (regions or objects) are labeled (typically, by a domain expert). Each image class is characterized by certain objects (e.g., body outline, liver, spine for MRI or CT images of the abdomen, skull, ventricles for images of the head etc.). Such objects are present in almost every image of the same class. A number of additional objects (unexpected) may also be identified and classified into one of a number of classes such as hematoma, tumor etc.

Figure 1 illustrates an example ARG representation for medical MRI images of the abdomen. 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) to the remaining objects. Both nodes and arcs are labeled by the attribute values of the region properties and the relationship properties, respectively. In this work we used the following set of features:

Individual regions: Individual objects are described by properties corresponding to characteristics of their position, size and shape. A set of features that has been used successfully [12, 7] is the following: *size* (s), computed as the size of the area it occupies, *perimeter* (p) computed as the length of its bounding contour, *roundness* (r), computed as the ratio of the smallest to the largest second moment and *orientation (o)*, 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 properties are used to describe the spatial relationships between regions: *relative distance* (rd) computed as the minimum distance between their surrounding contours, *relative orientation* (ro) defined as the angle with the horizontal direction of the line connecting the centers of mass of their regions and, *relative position* (rp), 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.

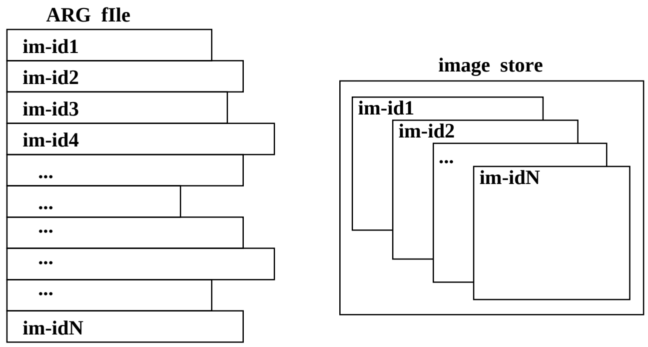

Figure 6: *File structure.*

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 their *sin* and *cos*. All attributes are normalized in the range $[0, 2]$. In particular, distances and areas are divided by the half of the perimeter and area of the largest region (i.e., body outline) respectively. Roundness takes values in $[0, 1]$ and is multiplied by 2. Relative distances are divided by the half of the maximum relative distance between any two regions. This normalization results into features which are scale invariant. These features are also translation invariant since only relationships between objects are used to characterize object positions. To achieve rotation invariance, all images are registered to a standard orientation (e.g., the axis of elongation of the outline object is made horizontal). Individual objects or regions are represented by 5-dimensional vectors of the form $(s, p, r, 1 + cos(o), 1 + sin(o))$ while, relative orientations are represented by 4-dimensional vectors of the form $(rd, 1 + rp, 1 + cos(ro), 1 + sin(ro))$.

Figure 6 illustrates the proposed file structure on the disk. It consists of the following parts:

The "ARG file". This is a file holding the ARGs. Each record in this file consists of (a) an identifier (e.g., image file name) corresponding to the image from which the ARG has been derived and (b) the features of each region together with its relationships with the other regions.

The "image store" holding the original image files. For faster display, we have also kept the segmented forms of all images.

All queries address the ARG file rather than the raw image data stored in the image store. We concentrate our attention to the case of queries by image example: A query image or sketch of dominant image segments is given, it is analyzed, its ARG is computed and compared with all stored ARGs. We adopted the Chebeychev L_∞ distance (Equation 4) for node and edge matching (i.e., edit operations). The ARG distance is computed by taking the maximum cost (Equation 2) of all the edit operations. The reason of our choice is that Chebeychev distance and the maximum yield low distance values only when all node and edge attributes in a query and in a stored image are similar. This will ensure that the most similar images to a query will be retrieved first. In addition, L_∞ and maximum compute faster than any other distance.

We implemented the system in C, on a dedicated SUN Ultra 1 running Solaris^{TM}. As a testbed, we used the dataset of $13,500$ synthetic segmented images[1] of [7]. To evaluate our system we created 25 characteristic queries. All images and the queries contain between 4 and 8 objects. Each query retrieves the best 50 answers. Retrievals took approximately 30 seconds on the average per query.

[1]The test data and the results are available from `http://www.ced.tuc.gr/~petrakis`.

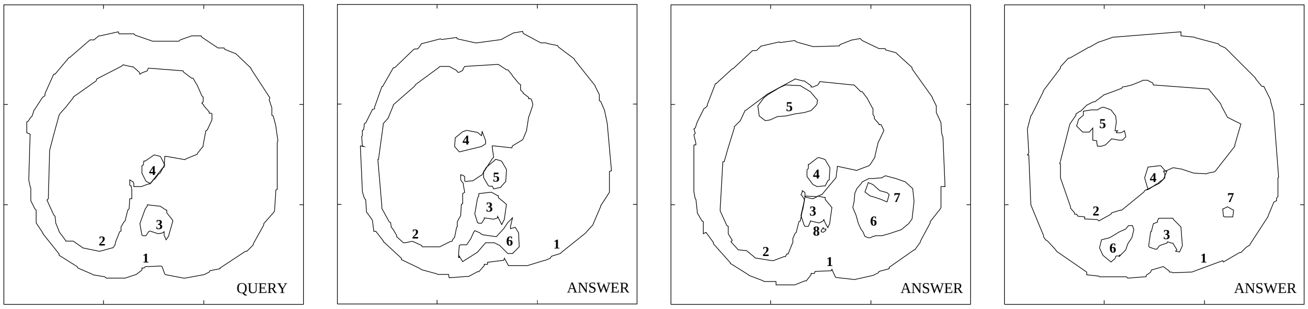

Figure 7: *Example of a query (left) and three retrieved images.*

Evaluations have been carried-out following the instructions of a radiologist: Two images (i.e., a query and a stored shape) are considered similar if they contain similar objects in similar spatial relationships.

number of retrieved images	1	5	10	20	30	40	50
precision	1.0	0.96	0.90	0.86	0.80	0.75	0.70

Table 1: *Average values of precision for retrievals by image content.*

To measure the accuracy of out method, for each query we computed *precision*, that is, the percentage of similar images retrieved with respect to the number of retrieved shapes. Table 1 illustrates the average values of precision as a function of the size of the answer from 1 up to 50. Each number in this table is the average over 25 queries. For small answer sets (i.e., with up to 5 answers) our method achieves precision higher than 0.95 that is, more than 95% of the retrieved answers (images) are correct. Figure 7 demonstrates a characteristic example of a query image (left) and 3 retrieved images. Objects 1, 2, 3 and 4 in the query image are matched with objects with the same indices in the retrieved images (for simplicity, image indices have been rearranged so that all associated objects have the same indices). Notice that, the retrieved images may contain extra objects.

5 Conclusions

An IDB system has been described which supports the efficient processing, archiving, and retrieval of medical images by content. The system consists of an interactive IDB environment, which supports the communication between the user and the various system components. Attention is focused on a methodology which supports the automated retrieval of medical images by content, based on both object properties and relationships between objects. The effectiveness of the proposed methodology has been assessed based on simulated but realistic tomographic images. The results are a good support to our claims of accuracy.

The system can be easily extended with additional features and mechanisms facilitating the processing and accessing of image data. For instance, the user interface may be extended with additional tools for image processing and registration, as well as with a powerful query language supporting various types of image queries, in addition to queries by example.

Acknowledgements

I would like to thank prof. Stelios Orphanoudakis for valuable contributions in this work.

References

[1] A. Gupta and R. Jain. Visual Information Retrieval. *Communications of the ACM*, 40(5):71–79, 1997. (http://www.virage.com).

[2] M. Flickner et. al. Query By Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, 1995. (http://wwwqbic.almaden.ibm.com).

[3] A. Pentland, R. W. Picard, and A. Sclaroff. Photobook: Content Based Manipulation of Image Databases. *Intern. Journal of Computer Vision*, 18(3):233–254, 1996. (http://vismod.www.media.mit.edu/vismod/demos/photobook/index.html).

[4] W. W. Chu, C.-C. Hsu, A. Cardenas, and R.K. Taira. Knowledge-Based Image Retrieval with Spatial and Temporal Constructs. *IEEE Trans. on Knowledge and Data Engineering*, 10(6):872–888, 1998.

[5] D. Comaniciou, P. Meer, and D. J. Foran. Image-Quided Decision Support System for Pathology. *Machine Vision and Applications*, 11(4):213–223, 1999.

[6] S. C. Orphanoudakis, C. Chronaki, and D. Vamvaka. I^{2}Cnet: Content-Based Similarity Search in Geographically Distributed Repositories of Medical Images. *Computerized Medical Imaging and Graphics*, 20(4):193–207, 1996.

[7] E. G.M. Petrakis and C. Faloutsos. Similarity Searching in Medical Image Databases. *IEEE Trans. on Knowledge and Data Engineering*, 9(3):i435–447, 1997.

[8] S.-K. Chang, E. Jungert, and G. Tortora, editors. *Intelligent Image DataBase Systems*. World Scientific, 1996.

[9] B. T. Messmer. *Efficient Graph Matching Algorithms*. PhD thesis, University of Bern, Switzerland, 1995. (http://iamwww.unibe.ch/˜ fkiwww/projects/GraphMatch.html).

[10] W. J. Christmas, J. Kittler, and M. Petrou. Structural Matching in Computer Vision Using Probabilistic Relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.

[11] P. Kofakis and S. C. Orphanoudakis. Image Indexing by Content. In M. Osteaux et. al., editor, *A Second Generation PACS Concept*, chapter 7, pages 250–293. Springer-Verlag, 1992.

[12] E. G.M. Petrakis and S. C. Orphanoudakis. Methodology for the Representation, Indexing and Retrieval of Images by Content. *Image and Vision Computing*, 11(8):504–521, 1993.