Tools and Methodologies for the Indexing, Storage and Retrieval of Medical Images[1]

Euripides G.M. Petrakis† and Stelios C. Orphanoudakis †‡

† Department of Computer Science, University of Crete
‡ Institute of Computer Science, FORTH
Heraklion, Crete, Greece

Abstract

The functional characteristics of a prototype Image DataBase system under development are presented and discussed. This system is based on the integration of tools and methodologies which support the interactive processing, classification and browsing of medical images, as well as methodologies for the efficient automated indexing, storage and retrieval of such images by content.

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 particular, such database systems may provide valuable clinical decision support and increase the efficiency of diagnostic image interpretations. Furthermore, such systems are emerging as an important component of Picture Archiving and Communications Systems (PACS) in order to support administrative, clinical, teaching and research activities.

An “Image DataBase” (IDB) is a “ system in which a large amount of image data and their related information are integratedly stored” [1]. Important considerations in the design and implementation of IDB systems are: image feature extraction, image content representation and organization of stored information, search and retrieval strategies, and user interface design. Traditional database concepts such as data independence, data integrity, and control of shared information are also of great significance. In the case of a medical IDB, image data include: the raw images themselves, attributes

[1] First International Workshop in Picture Archiving and Communication Systems (PACS), London, December 1992

(e.g., dates, names), text (e.g., diagnosis related text), information extracted from images by automated or computer assisted image analysis, modality and image file header information (e.g., ACR/NEMA) etc. Such data can be viewed as forming either “multimedia documents” [2], or “objects” [3].

The retrieval capabilities of an IDB must be embedded in its query language. On the other hand, query responses depend highly on query type, specificity, complexity, amount of on-line image analysis required and the size of the search. 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”: before a final selection is made, characteristic representations (e.g., icons, image miniatures) corresponding to all images contained in the answer set are displayed. Browsing can be especially helpful when the specification of pictorial content is ambiguous and it may be the only method for making a difficult final selection [4]. In addition, displaying such image forms instead of the original images themselves saves extensive data transfers through a network during retrievals.

Command oriented query languages allow the user to issue queries by conditional statements involving various image attributes (values of attributes and/or ranges of such values). Other types of image queries include: queries by identifier (a unique key is specified), region queries [5] (an image region is specified and all regions that intersect it are returned), text queries [6] etc. The highest complexity of image queries is encountered in queries by example. In this case, a sample image or sketch is provided and the system must analyze it, extract an appropriate representation of its content and, finally, match this representation against representations of images stored in the database.

The effectiveness of an IDB system supporting the automated archiving and retrieval of images by content can be significantly enhanced by incorporating into the IDB storage and search mechanisms efficient techniques supporting the automated indexing and retrieval of images by content (i.e., in terms of properties of both the objects contained in images and the relationships among them). So far, in order to determine which images must be retrieved, content representations corresponding to all stored images are compared (one by one) with a similar representation extracted from the query image. However, retrievals may become inefficient, since comparisons may involve time intensive operations such as graph matching [7, 8]. Various other techniques with lower time complexity, such as the “2D string” matching technique [9], can be used to resolve such queries.

Two are the main contributions of this work towards developing an IDB system which supports the efficient processing, archiving and retrieval of medical images by content. First, an environment has been developed consisting of a user interface and graphical tools facilitating the interaction between the user and the various system components. For example, in order to deal with the problems of efficiency, uncertainty and complexity in determining the content of images, the user is allowed to interact with the IDB and correct the results of image segmentation. Furthermore, the user is allowed to specify the class to which an image belongs.

Once the content of images has been extracted reliably, the system may resume responsibility for the efficient representation, storage and retrieval of images. The second contribution of this work is the development of a new methodology which supports the efficient processing of queries by image example as well as the efficient representation, indexing and retrieval of images by content. In particular, images may be indexed and accessed based on spatial relationships between objects and properties of individual objects. Such tools and methodologies will be combined and integrated into a prototype IDB system, which is currently under development in our laboratory.

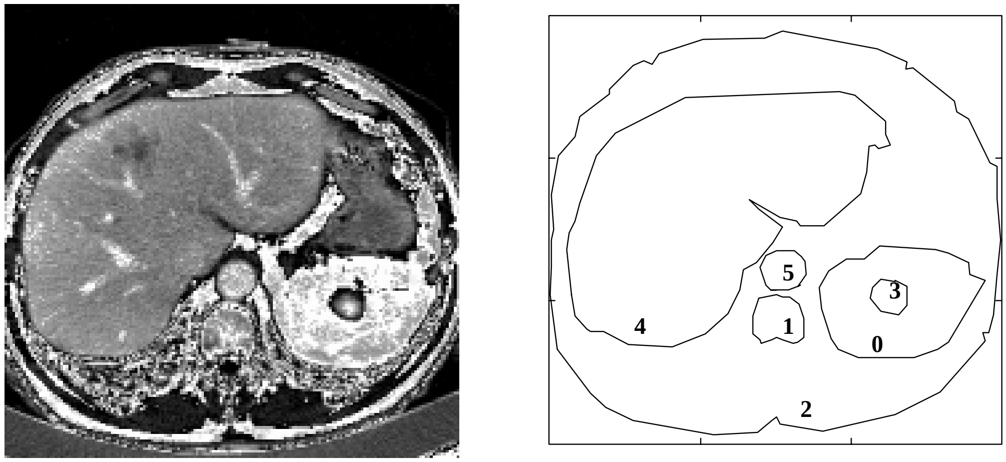

Figure 1: Example of original grey level image (left) and its segmented form (right).

2 An Interactive IDB Environment

An environment has been developed and implemented which supports the interactive processing, classification and browsing of medical images (CT and MRI images of the body) [10, 11]. Such an environment can be used over an existing IDB system for the purpose of assisting the interaction between the user and the IDB. In particular, this environment supports the:

• **Processing and segmentation of medical images:** before any required image descriptions are extracted and used (e.g., stored or matched with others stored in the IDB), images must first be segmented into disjoint regions or objects. The segmentation of CT and MRI images is in general very difficult and it is currently the subject of independent research activities [12, 13]. Thus, segmentations are often carried out under supervision. First, images are segmented using low-pass filtering followed by edge detection [14]. A polygonal approximation to the derived segment edges is then obtained. The desired segmentation results are then obtained by editing (i.e., the user 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. Figure 1 shows an example of an original grey-level image (left) and its corresponding final segmented polygonal form (right).

• **Classification of medical 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 a part of an image (i.e., an object, segment or 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 2 shows an anatomical hierarchy [11]. The higher level of abstraction is on the left and the lower level is on the right.

• **Efficient IDB organization:** An image or segment may be simultaneously an instance of more than one classes found at any level in a particular hierarchy. The instances of each class may be

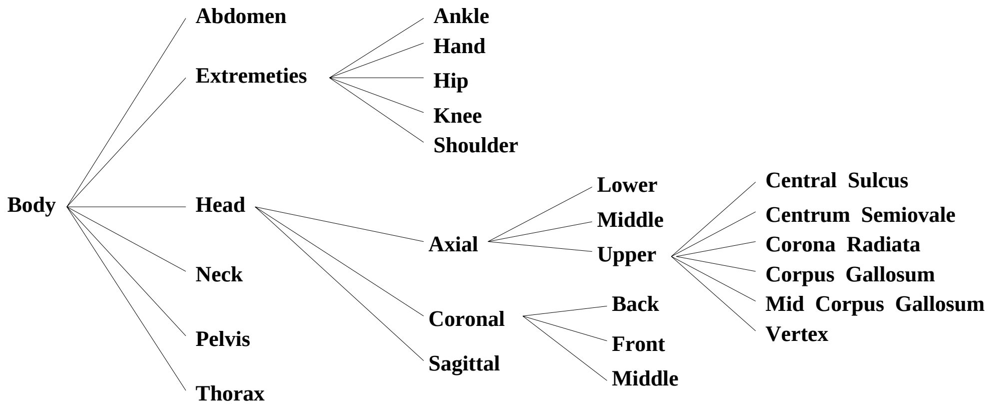

Figure 2: Anatomy hierarchy defined.

stored (logically or physically) separately, so that the IDB is partitioned into database segments. This reduces the search space leading to faster retrievals, since a query addresses only a specific database segment. Higher level classes inherit the instances of all their descendant (lower level) classes.

• **Retrieval of medical images:** queries may be specified (a) by a conditional statement involving various attributes and (b) by specifying an example image (grey level image or sketch). A user may interactively specify the specific database segment to be searched. All images satisfying query criteria need to be retrieved, while a user is allowed to make a final selection by browsing. Images and image related data (e.g., text, attributes) stored in the database may both need to be retrieved in response to a particular query.

• **Browsing of the IDB:** The user interface is equipped with interactive graphical tools and mechanisms for both displaying and selecting image classes or class properties and for exploring their interrelationships. Such display tools are referred to as "graph browsers" in [10]. Furthermore, the user interface is equipped with an "image browser" whose purpose is to display reduced resolution images (miniatures) or sketches corresponding to the images stored in the database.

3 Image Representation, Indexing, Storage and Retrieval

A new methodology for the efficient representation, indexing, storage and retrieval of two-dimensional images by content is presented and discussed. Indexing can be based on certain types of spatial relationships such as: "left/right", "below/above" and "inside/outside". Images can also be indexed in terms of spatial properties specific to individual objects such as size (area), roundness (elongation), orientation with respect to a reference direction, or in terms of properties of object classes. Such image properties have the advantage of being generally useful for many kinds of images and imaging applications.

Prior to storage, images are decomposed into groups of objects, called "image subsets", each containing an equal number of objects. From an image containing n objects, all image subsets containing k objects are produced, with k ranging from 2 up to a prespecified number K_{max}. For instance, the image of Figure 1 which contains 6 objects numbered 0 through 6, gives rise to 15 image subsets of length 2, 20 subsets of length 3, 15 subsets of length 4, 6 subsets of length 5 and 1 subset of length 6 (which is the original image itself).

An image subset is considered to be the basic entity in the proposed indexing scheme. In particular, images are indexed based on representations of the set of all the derived subsets. The objects contained in each group are first ordered. Ordering must be based on criteria that clearly differentiate the objects among them. Position is such a criterion, since objects are usually scattered in the image. Size or shape do not necessarily provide good ordering criteria, since images may contain similar objects. Each of these ordered subsets is then represented by a set of attribute strings corresponding to the set of properties involved in a particular image representation. An individual image subset is indexed by treating each of its corresponding attribute strings as a separate key.

3.1 Indexing Based on "Left/Right" and "Below/Above" Relationships

Let $(a_0, a_1, \ldots, a_{k-1})$, be an image subset of length $k \in [2, K_{max}]$. For any two objects a_i, a_j, $i, j \in [0, k-1]$, with centers of gravity (x_i, y_i) and (x_j, y_j) respectively, either a_i is a "predecessor" of a_j which is written as: $a_i \prec a_j$, or a_i is a "successor" of a_j which is written as: $a_i \succ a_j$. Specifically,

$$\forall\, i, j \;\in\; [0, k-1] \quad \begin{cases} a_i \prec a_j, & \text{if } x_i < x_j \text{ or, } y_i < y_j \text{ if } x_i = x_j; \\ a_i \succ a_j, & \text{otherwise.} \end{cases} \tag{1}$$

The application of this ordering criterion to the objects contained in an image subset gives rise to a permutation string p, which is actually the ordered sequence of indices corresponding to the above objects. String p corresponds to the sequence of objects produced by projecting their positions along the x axis and by taking them from left to right. For instance, the permutation string p corresponding to the image subset (2 3 4 5) is (4 2 5 3). Similarly, a second permutation string $p^{'}$ is produced by projecting the positions of objects along the y axis and by taking them from below to above. In particular, string $p^{'}$ is produced by ordering objects according to the following rule:

$$\forall\, i, j \;\in\; [0, k-1] \quad \begin{cases} a_i \prec a_j, & \text{if } y_i < y_j \text{ or, } x_i < x_j \text{ if } y_i = y_j; \\ a_i \succ a_j, & \text{otherwise.} \end{cases} \tag{2}$$

The permutation string p' corresponding to the image subset (2 3 4 5) is (3 5 2 4). The rank r_i of object p_i with respect to the sequence p' is defined as the number of objects preceding it along the y direction. Specifically, r_i is computed as follows:

$$r_i = j \quad \Longleftrightarrow \quad p_i = p'_j, \quad 0 \le i, j < k. \tag{3}$$

A string r is produced, corresponding to the ordered sequence p. The rank string r corresponding to the ordered image subset (4 2 5 3) is (3 2 1 0). The two-dimensional string (p, r) will be used henceforth to represent the "left/right" and the "below/above" relationships between objects contained in a given image subset. In particular, given a representation (p, r), the "left/right" and "below/above" relationships between the objects contained in image subset p can be determined as follows:

$$\forall\, i, j \;\in\; [0, k-1] \quad \begin{cases} \text{object } p_i \text{ is on the left of } p_j, & \text{if } i < j; \\ \text{object } p_i \text{ is on the right of } p_j, & \text{if } i > j. \end{cases} \tag{4}$$

Similarly:

$$\forall\, i, j \;\in\; [0, k-1] \quad \begin{cases} \text{object } p_i \text{ is below } p_j, & \text{if } r_i < r_j; \\ \text{object } p_i \text{ is above } p_j, & \text{if } r_i > r_j. \end{cases} \tag{5}$$

Such representations are both translation and scale invariant: images translated or scaled with respect to each other result in the same representation. Translation invariance is assured, since only relative positions are taken into account in determining the order of objects. Similarly, scale invariance is assured, since no distance criterion is used. On the other hand, resulting representations are rather sensitive to image rotations. Rotation invariance can be achieved only in cases where a reference direction can be identified (e.g., specified interactively by the user) so that, prior to any processing, images are rotated and placed to a standard orientation.

3.2 Indexing Based on the Inclusion Relationships

Given an ordered image subset $p = (p_0, p_1, \ldots p_{k-1})$, a string $w = (w_0, w_1, \ldots w_{k-1})$ representing the "inside/outside" relationships among the objects it contains is constructed as follows:

$$w_i = \begin{cases} j & \text{if object } p_i \text{ is both closer and inside object } p_j, \quad 0 \leq i, j < k; \\ i & \text{if object } p_i \text{ is contained by the image frame only.} \end{cases} \tag{6}$$

An object is always contained by the image frame. The distance between any two objects is defined as the minimum distance between their bounding contours. For instance, the inclusion string w corresponding to the ordered image subset (4 2 5 3) is (1 1 1 1).

3.3 Indexing Based on Properties of Objects

The description of individual objects is given in terms of properties corresponding to global object characteristics such as: area, perimeter, roundness, orientation with respect to a reference direction and/or by features that are shown to be effective in quickly narrowing down the search space for the purpose of image retrieval by content in a particular application domain. Given an image subset p, each of the above properties gives rise to a separate attribute string $(a_0, a_1, \ldots a_{k-1})$, where:

$$a_i = \left\lfloor \frac{\text{property value of object } p_i}{\text{maximum property value}} \right\rfloor \cdot q - 1, \quad 0 \leq i < k. \tag{7}$$

The maximum property value is usually different for different properties. Roundness has maximum value 1 (corresponding to a circle), orientation has maximum value $\pi = 3.141\ldots$, etc. The resulting representation must be independent of scale. Therefore, maximum size (e.g., perimeter, area) values are set equal to the size of the biggest object in p. q is an integer, called "quantization value", corresponding to the number of different values a property may take. For example, if for some property $q = 3$, then an object is assigned an integer property value in the range $[0, 2]$. The string encoding the size (area) property of the objects contained in the ordered image subset (4 2 5 3) is (1 2 0 0), while the string encoding the roundness property is (1 2 2 2).

3.4 Image Storage

Image data can be distinguished into "physical" and "logical" [15]. Original (grey level) images, segmented images, and image miniatures are physical images. On the other hand, image related data (information extracted from images, attributes, text etc.) are logical images. Physical and logical images are stored separately in a physical and a logical database respectively. Pointers are implemented from the logical to the physical database. To reduce storage requirements, physical images are compressed prior to storage and conversely decompressed upon retrievals. Todate, no image file structure has been introduced. However, images will eventually be stored in clusters based on the likelihood of being retrieved together in response to a particular query (e.g., the set of all images corresponding to a patient's exam may be stored close together on disc).

The logical database consists of a set $(H_2 \ldots, H_{K_{max}})$ of relational tables, where K_{max} is the maximum size of image subsets under consideration. The specification of value K_{max} depends on the application. The purpose of each H_k table is to resolve queries which specify k objects. Therefore, K_{max} can be set equal to the maximum number of objects allowed in queries, if such a value can be specified in advance. In general, K_{max} may take any value greater than or equal to 2. Typically, the number of objects specified in image queries is not greater than 6. Therefore, we set $K_{max} = 6$.

The image subsets of size $k,\ 2 \leq k \leq K_{max}$, together with their representations are all stored in table H_k. Each image subset is represented by a tuple of one dimensional strings of the form: $(p, r, w, \ldots)$ where p is the ordered sequence of object indices, r is its corresponding rank string representing the "left/right" and "below/above" relationships between objects, w is the inclusion string, and the remaining strings correspond to properties of individual objects. Two such strings have been used, s to encode the size property and c to encode the roundness property of those objects whose indices are in p. Therefore, the representation of an image subset is given a tuple of the form (r, w, s, c). Indexing is performed by creating a secondary index for each attribute string.

In general, the logical database may be considered as consisting of a set $(H, H_1, H_2 \ldots, H_{K_{max}})$ of relational tables. Table H stores information about each image as a whole and has attributes such as: image file name, dates, names, text descriptions, image header information etc. Table H_1 on the other hand, can be used to store representations of the shape of objects such as those proposed in [16, 17, 18, 19].

3.5 Image Retrieval

All queries address the logical database rather than the raw image data stored in the physical database. 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, and a representation similar to that of the images stored in the IDB is created. All images containing at least one image subset matching the query (have both equal size and same property strings) are then retrieved and displayed for viewing. Image subsets of equal size having the same property strings with those of the query are also retrieved and displayed. Query processing depends on the number of objects m contained in the query image. In particular, we distinguish between, "direct access" queries corresponding to $2 \leq m \leq K_{max}$ and "indirect access" queries corresponding to $m > K_{max}$, where K_{max} is the maximum size of image subsets stored.

3.5.1 Direct Access Queries: $2 \leq m \leq K_{max}$

The representation (p, r, w, s, c) of the query is derived first. Then, the query addresses the H_m table and all subsets with the same representation are retrieved. Retrieved image subsets may be also compared on the basis of additional properties such as properties relating to the shape of the objects they contain (stored in table H_1).

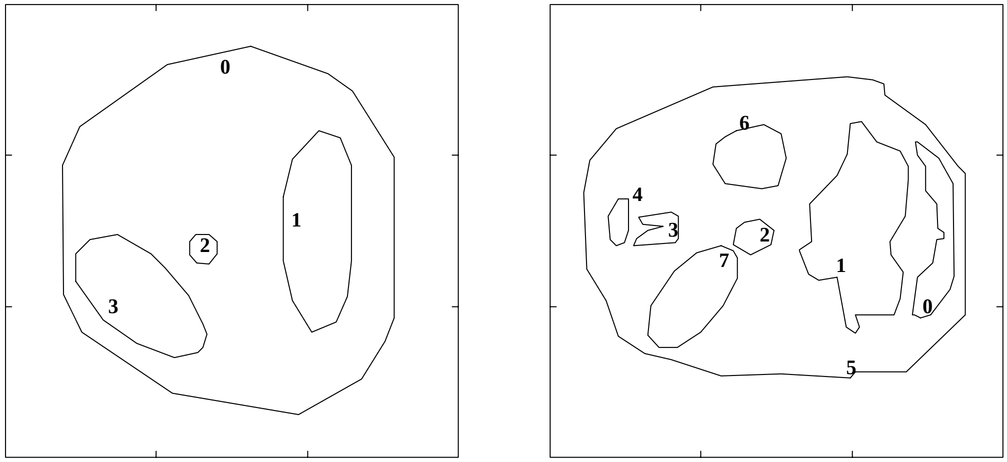

Figure 3: Example of query (left) and retrieved image (right).

3.5.2 Indirect Access Queries: $m > K_{max}$

Query processing consists of two steps. During the first step (hypothesis generation), a set of candidate image subsets is retrieved: among the m objects contained in the query image, a set of K_{max} objects is taken at random, thus creating a new query. All image subsets having the same representation with the representation of the new query are retrieved. During the second step (hypothesis verification), all image subsets obtained from images corresponding to hypothesized image subsets are generated and matched with corresponding representations of the original query. Figure 3 shows an example of a query image containing 4 objects (left) and an example of a retrieved image (right). The representation obtained from the query image is $(p, r, w, s, c) =$ (3 2 0 1, 0 1 2 3, 2 2 2 2, 0 0 2 0, 1 2 2 0). The image subsets matching the query are (7 2 5 0) and (7 2 5 1). Query object 3 matches object 7 of the retrieved image with respect to the size and roundness properties but not with respect to orientation, since the property of orientation is not among the properties matched. The quantization parameter q was set equal to 3.

3.6 Performance Evaluation

Evaluations have been carried out based on a number of test queries addressing a prototype IDB storing 223 medical CT and MRI images of the human body. Queries are distinguished based on the number m of objects they specify. To obtain average performance measures, for each value of m ranging from 2 to 6, 16 image queries have been used and the average performance to queries specifying an equal number of objects has been computed. Measurements of both the answer set (percentage of images returned with respect to the total number of images stored) and of the retrieval response times have been taken and shown in Figure 4. Queries become more specific and the size of an answer set decreases as the number of objects contained in queries increases. Therefore, response time decreases with the number of query objects since search space and thus the amount of information to be processed decreases too.

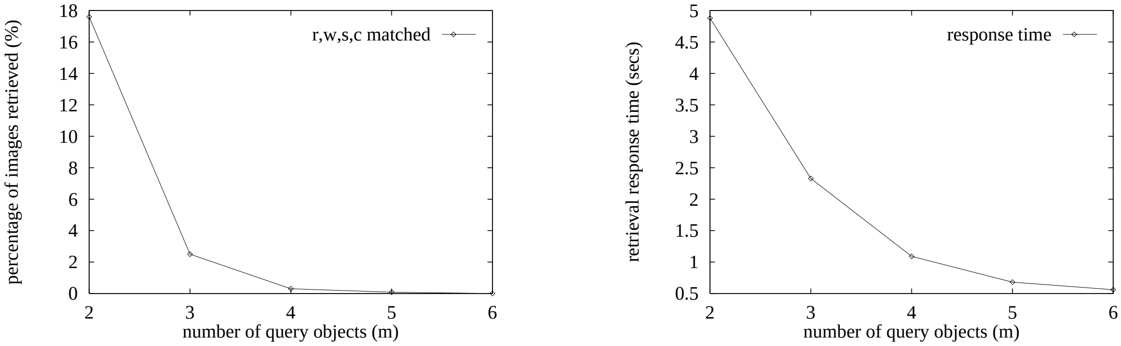

Figure 4: Average size of an answer set with respect to the number of query objects as a percentage of images returned (left) and corresponding average retrieval response times in secs (right).

4 Discussion

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, and a methodology which supports the automated indexing and retrieval of images. Such tools and methodologies will be integrated into a prototype IDB system which is currently under development.

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 and mechanisms for image registration and image processing, as well as a powerful query language supporting various types of image queries, in addition to queries by example (e.g., queries by specifying an identifier, a class, a hierarchy, range queries etc.), such as those proposed in [11]. Furthermore, the proposed methodology for image indexing can be easily extended to include the indexing of image sequencies in three or four dimensions, where the fourth dimension is time.

In developing an IDB system which supports the efficient management of various kinds of image data, is extensible and easy to use, we have to choose an appropriate data model and a database management system (DBMS) which provides persistent storage of both the model and the data, as well as mechanisms for defining, creating, modifying and accessing the model and the data. Furthermore, it must provide a query language, transaction management, concurrency control and authorization for a multiuser environment, as well as performance features such as secondary indexing and clustering. Such features and mechanisms are inherent within a relational DBMS. Besides, a relational DBMS offers the most direct way of implementing the logical database proposed in this work.

However, in order for a DBMS to be appropriate for IDB work it must be augmented with semantic data modeling concepts (e.g., class definition and hierarchies) to assist application modeling. In particular, in developing an IDB system which satisfies the need for a hierarchical database organization, takes advantage of the property of inheritance, and is extensible, the object-oriented approach [3] seems to be more appropriate. All database entities (e.g., various types of data) will be defined as either primitive or complex objects, while system functions (e.g., image processing and retrieval functions) will be defined as methods encapsulated within the same representation with the above database entities. Extensions of relational data models and DBMS's with object-oriented characteristics do exist and can be used to develop an IDB system which satisfies our needs. POSTGRESS [20] is an example of such a model and system.

Acknowledgements

We would like to thank Dr. P. Kofakis, Y. Kavaklis, S. Kostomanolakis and D. Lagouvardos for valuable contributions in this work.

References

[1] Hideyuki Tamura and Naokazu Yokoya. Image Database Systems: A Survey. *Pattern Recognition*, 17(1):29–49, 1984.

[2] D. Tsichritzis et.al. A Multimedia Office Filing System. In *Proc. 9th International Conference on Very Large Data Bases*, 1983.

[3] Won Kim. Object Oriented Databases: Definitions and Research Directions. *IEEE Transactions on Knowledge and Data Engineering*, 2(3):327–341, 1990.

[4] Panos Constantopoulos, Stelios C. Orphanoudakis, and Euripides G. Petrakis. An Approach to Multimedia Document Retrieval on the Basis of Pictorial Content. Technical Report 011, Institute of Computer Science, Foundation of Research and Technology - Hellas, Heraklion, Greece, February 1988.

[5] Christos Faloutsos and Yi Rong. Spatial Access Methods Using Fractals: Algorithms and Performance Evaluation. Technical Report UMIACS-TR-89-31, CS-TR-2214, University of Maryland, Colledge Park, Maryland, February 1989.

[6] Christos Faloutsos. Access Methods for Text. *ACM Computing Surveys*, 17(1):49–74, March 1985.

[7] Martin A. Fischler and Robert A. Elschlager. The Representation and Matching of Pictorial Structures. *IEEE Trans. on Computers*, c-22(1):67–92, 1973.

[8] Linda G. Shapiro and Robert M. Haralick. Structural Discriptions and Inexact Matching. *IEEE Trans. on Patt. Anal. and Mach. Intel.*, 3(5):504–519, 1981.

[9] Shi-Kuo Chang, Qing-Yun Shi, and Cheng-Wen Yan. Iconic Indexing by 2-D Strings. *IEEE Trans. on Patt. Anal. and Mach. Intel.*, 9(3):413–428, May 1987.

[10] Petros Kofakis and Stelios C. Orphanoudakis. Graphical Tools and Retrieval Strategies for Image Indexing by Content. In *Proceedings of Computer Assisted Radiology, CAR91*, pages 519–524, Berlin, July 1991.

[11] Petros Kofakis and Stelios C. Orphanoudakis. Image Indexing by Content. In M. Osteaux et. al., editor, *Hospital Integrated Picture Archiving and Communication Systems*, chapter 7, pages 265–309. Springer-Verlag, 1992.

[12] Gianni L. Vernazza, Sebastiano B. Serpico, and Silvana G. Dellepiane. A Knowledge-Based System for Biomedical Image Processing and Recognition. *IEEE Transactions on Circuits and Systems*, Cas-34(11):1399–1416, 1987.

[13] Ioannis Kapouleas. Segmentation and Feature Extraction for Magnetic Resonance Brain Image Analysis. In *Proceedings of 10th International Conference on Pattern Recognition*, pages 583–590, Atlantic City, New Jersey, June 1990.

[14] Dana H. Ballard and Christopher M. Brown. *Computer Vision*, chapter 4, pages 119–149. Prentice Hall, 1982.

[15] Shi-Kuo Chang. *Principles of Pictorial Information Systems Design*, chapter 8, pages 172–211. Prentice Hall International Editions, Englewood Cliffs, New Jersey 07632, 1989.

[16] Wen-Hsiang Tsai and Shiaw-Shian Yu. Attributed String Matching with Merging for Shape Recognition. *IEEE Trans. on Patt. Anal. and Mach. Intel.*, 7(4):453–462, 1985.

[17] Alan Kalvin, Edith Schonberg, Jacob T. Schwartz, and Micha Sharir. Two-Dimensional, Model-Based, Boundary Matching Using Footprints. *The International Journal of Robotics Research*, 5(4):38–55, 1986.

[18] Fridtjof Stein and Gerard Medioni. Efficient Two Dimensional Object Recognition. In *In Proccedings of 10th International Conference On Pattern Recognition*, pages 13–17, Atlantic City, New Jersey, June 1990.

[19] George Bebis, George Papadourakis, and Stelios C. Orphanoudakis. Model Based Object Recognition Using Multiresolution Segmentation and Artificial Neural Networks. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991.

[20] L. Rowe and M. Stonebraker. The POSTGRESS Data Model. In *Proceedings of the 13th Int. Conf. on Very Large Databases*, pages 83–95, Brighton, England, September 1987.