Content-Based Medical Image Retrieval in Object Oriented Databases

Caetano Traina Jr.[1]; Agma J. M. Traina[1]; Rildo R. dos Santos[2]; Edna Y. Senzako[1];
[1]Computer Science Department - Mathematics Institute at Sao Carlos of
Sao Paulo University - USP - Brazil
[2]Computer Science Dept - State of Sao Paulo University at S.J. Rio Preto - UNESP - Brazil
E-mail: [Agma|Caetano|Senzako]@icmsc.sc.usp.br , Rildo@dcce.ibilce.unesp.br

Abstract

This work describes the concepts used in an Object Manager aimed at storing medical images as one more data type associated to objects. Images can be retrieved through queries based on the graphic contents of the stored images. The usual approach uses icons and textual attributes stored with the images to specify the queries. This work uses a novel modeling technique to define the "image data type," through which it is possible to decide, beforehand of the query itself, the valuable data of each image that must be extracted from the image when it is stored in the database, so the search can be accelerated when queries are issued.

1. Introduction

Picture Archiving Computer Systems (PACS), applied in medical image storage, provide graphical information of the many aspects of the health, diseases and treatment of each patient for the physician and technicians of a health care center [1]. Today medical exams can generate a large amount of data due to the computerized medical instruments used to collect the graphical information about the patients [2].

Existing systems usually store descriptive information associated with the images, such as the comments and diagnoses already made by a physician, or shortened icons of each image. To find images through textual data, a detailed description is obtained with the help of a specialist, but the description depends on his/her knowledge of each specific topic, or on his/her purpose at the time of analysis. Therefore, textual data cannot substitute the actual information contained in an image. An iconic image is also not appropriate for the domain of a medical image database, because it is difficult to select an icon that could, in a significant way, represent the complex information present in this image [3] [4]. So, meta-information cannot substitute the actual information contained in an image.

The search for stored images based on their full contents, although highly desirable, imposes a processing load on the search data engine that cannot be accomplished within an acceptable timeframe in the current machines. This work describes an alternative way to deal with this excessive work load by pre-processing each image as it is being (fully) stored into an Object Oriented Database Manager (Object Manager), extracting a set of parameters that can be used to construct a set of index structures. The use of index structures can quickly reduce the target set of images to be retrieved when queries are issued.

This work is being implemented using an Object Manager [5] and an Image Matching System [6], which supports the concept of images as a new data type. Applications can define "objects" whose attributes can be of type integer, text, images, and so on. For instance, an object of the type person can have an attribute "age"of a type *integer* and another attribute "lung x-ray"of the type *set of images*. By using an Object Manager, it becomes possible to store complex attributes, such as images and methods, associated with the objects, in a straightforward manner, as will be described in sections 2 and 3. Section 4 shows how this system can be configured to meet the needs of an application or a particular operational environment.

2. Concepts and language elements

Images are defined in the Object Manager as part of its data meta-model, as a specialization of the concept of Attributes associated to Objects. The meta-model of this system assumes every attribute has a **characteristic**, which defines the set of operations in which it is involved. For example, a numbers is an attribute characteristic, so numbers are attributes that can be summed, multiplied, and so on; strings are another attribute characteristic, and strings can be searched for substring matching, can be concatenated, and so on. It must be noted that the set of operations allowable over an attribute characteristic is independent of the exact data type used to represent that attribute, so it does not matter if a number is represented as an integer, a real or a double precision float. Characteristics are specializations of the concept of object attribute that are further specialized in the Data Type concept, so each data type can have a specific set of properties. For example, complex numbers can have a method to calculate their real part. Images and procedures (methods) are two other attribute characteristics of this system, enabling various image representation formats (image data types) and image processing algorithms to be associated to objects like any other numerical or textual attributes.

To support the definition of images in this meta-model, the following concepts are needed [7]: **Image Constantss**, **Image Summarizer**, and **Image Parameters**. *Image Constants* are predefined images, meaningful in the application domain. For instance, a physician could have obtained a tomography where a given organ pathology appears in a predefined manner, that could be used as a reference to search or to identify other images with similar visual information. This image is, thus, ascertained as an Image Constant, so that it will be used by the system as a comparison pattern.

An *Image summarizer* is defined as a procedure that operates over two images, extracting useful data from the image, called *Image Parameters*. Image Parameters can be a numerical value, an image coordinate, or a set of values, etc. Each Image Summarizer can return one or more Image Parameters, each of them representing some useful information for the process of image selection and retrieval.

Another construction was included to yield the many variations in which a target object can appear in an image. That construction is called a **modifier**, and its purpose is to allow alterations on the default action of each image summarizer. The acknowledged variations in the images do not affect the basic behavior of the summarizer, yielding image variations like geometric transformations, contrast, etc. The application of any of these modifiers is directed toward two categories of constant image information that must be retrieved: i) *Partial Constant Image*, means the comparison of one target image with the image constant can be found in sub-images of the image, and ii) *Full Constant Image*, in which each image constant must be entirely compared with the target image.

Using these concepts, images can be stored associated to objects in the database, and are treated as a characteristic of object attributes of a given data type. However, to speed up the queries involving image content data, the operational environment must be configured for the specific application domain. This is achieved defining the set of image constants and image summarizers that are meaningful for each application. The idea is that, when an image is to be stored in the database, it is submitted to a predefined set of <image summarizer, image constants> pairs, and the resultant image parameters are used to create index structures, representing some form of "distance" of each stored image with this Image Constant.

When an application is built, the set of Image Constants related to each summarizer is defined, as well as over which object attribute type the summarizer needs to be evaluated. Thus, when each image is stored in the database as an attribute value, only the pairs <summarizer, constant> related to this particular association of the image is evaluated. Each of the resulting parameters is used to create the index structures, one for each image summarizer applied over each image constant meaningful to this object attribute. Each index structure is a modified R-tree [8], representing the image in a N-ary space, where N is the number of dimensions of the particular parameter. In the retrieval environment, each parameter is attached to a user defined range, so each query retrieves the images that each parameter has within the corresponding range. Each image can have indexes in many trees, one for each parameter of each summarizer meaningful for the object attribute type from which the image is a value.

In medical applications it is very important to work with adjustable matching methods, because images do not need to match exactly with an image constant to be in the result set of a specific query. For instance, images that match 85% or more of the image constant can be considered a good answer, and they must be included in the index structure. Methods have been implemented that permit retrieval of images that match exactly with a stored image, as well as methods where we can trim the degree of similarity accepted as a match. The matching methods implemented are the Correlation Coefficient, Correspondent Point Summations, Sequential Similarity Detection Algorithm-SSDA [9], Moment-Preserving Pattern Matching-MPPM [10].

3. The Image Content Schema

The application schema shows a semantic description of the information contained in each instance of an object attribute of a type image. The application domain is defined using objects of the type Image Constant, each having the attributes: *name*; *type* (partial or full Image Constant); and others describing the image. The characteristics that can be extracted from each stored image, through some comparison with one Image Constant, are represented as an object of the Summarizer type, whose attributes are: *name*; *type* - meaning the image processing algorithm (histogram evaluation, matching searches, texture identification, etc.); and *modification* - an indication that the summarizer can have another definition for its default action. Each parameter returned by some Image Summarizer is represented as an object of the Description_Variable type. Its attributes are: *name*; *type* of information

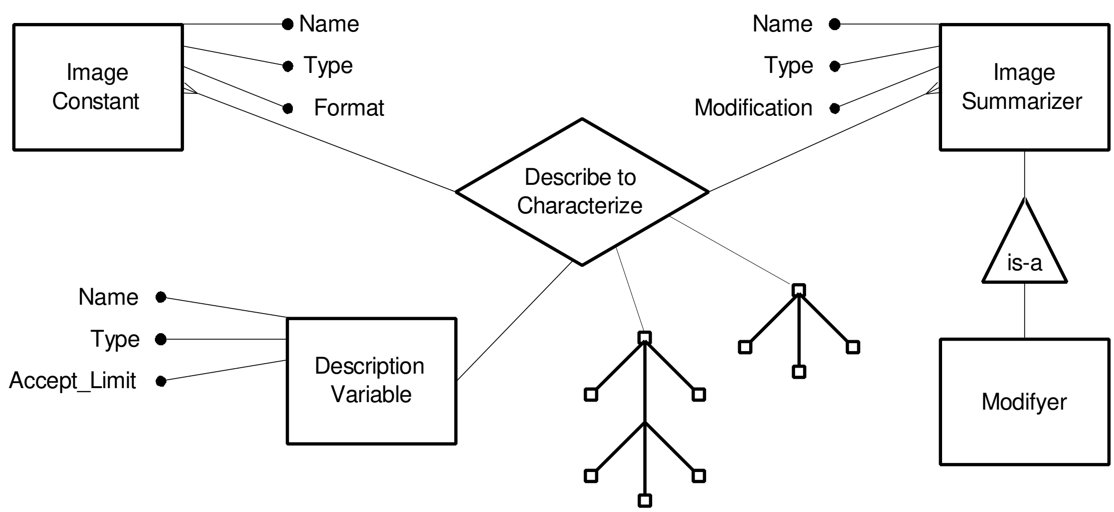

Figure 1: Meta-schema of an image supporting application.

returned by the Summarizer; and an *Accept_Limit* defining the threshold from which some image will have one entry into the index structure. The Summarizer Modifiers are specializations (subtypes) of Summarizers, and thus are in fact alternative image processing algorithms used in particular situations, depending on the query, or on the object to which the image is attributed.

Figure 1 shows an overall diagram of the Meta-schema of this structure using the notation of the Entity-Relationship Data Model. The whole information extracted from one image when it is stored in the data base is represented by a “Describe_to_Characterize” ternary relationship between objects of the Image Constant, Summarizer and Description Variable types, as shown in the schema in figure 1. The interpretation of this relationship provides a semantic meaning for each value stored in the index structures. The index structures are accessed from that relationship. The image itself is represented as an Attribute specialized to be an image, whose own set of attributes consists of: *Attribute_type_Name*, the name given to this type of attribute in the application schema; *Instance_Name*, an identifier of each image instance in the data base. The other attributes are specific for this instance, such as *length*, the *image* itself (the pixel matrix), *dimX*, *dimY*, and its *format*.

4. Operational Environment

The system described in this work presupposes the existence of three user groups: the developers, the production group and the client group. Developers are those responsible for the analysis and implementation of the application software inside a given application domain. They also develop the image processing algorithms, defining the Image Summarizers and the associated parameters. These algorithms are implemented as methods, and associated to objects in the database as attributes of the characteristic “procedure”. The applicability of these Image Processing Methods (IPM) depends on the object to which it is associated. Assigning an IPM to a user object restricts its use solely to this object. Assigning an IPM to a meta-object of the Image Summarizer type defines it as an image summarizer.

The production group consists of the users that will configure the application and the image database, trimming them for use by the client group. They will define the Image Constants and the image processing methods meaningful to use as a summarizer for each object attribute of a data type image defined in the application schema. This group is subdivided into: a) Image Database Manager - the person who works in collaboration with the developers and the medical staff to define the application schema, following the concepts presented in the image meta-schema to define the Image Summarizers that must be implemented and the associations between pairs <summarizer, constant> involving each summarizer, and each object attribute of a data type image; and b) the Application Managers, who work in collaboration with the Image Database Manager to define the image retrieval and manipulation plans, through predefined macros, to meet the needs of predefined queries already known.

The Client Group is composed of physicians and end users, the people who use the application and the image retrieval and manipulation plans to query the system. By allowing the queries to be executed through predefined macros, the application manager can define the index structures that must be maintained by the Image Database Manager, who in turn dictates those summarizers the developers must supply.

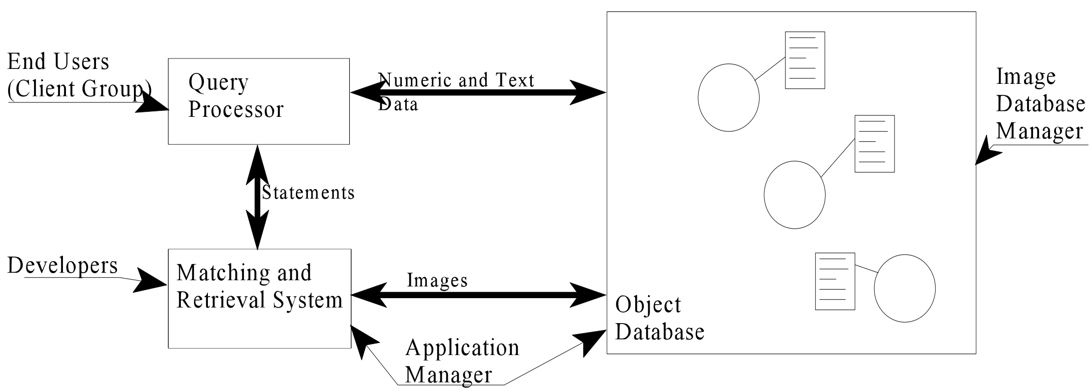

Figure 2: Arquitecture of the whole Image Content-based Retrieval System.

When a user sends queries to the database through the application, the predefined macros select a set of prospective images, narrowing the set of images that must effectively be processed, drastically reducing the processing time needed by the query. This technique induces the construction of expansible systems, where new Image Processing Algorithms can be dynamically appended into an application by the Image Database Manager, without code modifications. This can be done by linking the new Algorithm with its meaningful image constants through the attribute types. The results from the algorithm execution create one new index structure for each of its parameters, whose traversing can be embedded in the macros generated by the Application Manager. Figure 2 shows the architecture of the whole system, and the point where each user interacts with it.

5. Conclusion

This work describes the use of the concept of images as a data type, and Image Constants and Image Summarizers as a way to pre-process images and speed up the content-based retrieval of images stored in an Image Database. The concepts presented herein can be used in cooperation with more traditional searches based on descriptive text strings and icons attached to the images, although the described method can almost always replace the others. The proposed approach keeps the search precision by storing the entire image. It accelerates image retrieval processing by distributing the workload of the image processing methods in the storing time. Besides take advantage of the time when the workload is slower, the results of the processing are stored so each computation is done only once. Querying against even a large set of images is fast, since at the time of the search much of the computation has already been done.

This approach can use a set of indexes to quickly discard a lot of images that undoubtedly are not part of the response set of a query, enabling submission of only a reduced set of images to the more time-consuming filters needed to complete the answer.

The fact that modifications in an image by a given image processing algorithm correspond to equivalent changes in the image parameters returned by an image summarizer, permits manipulation of the index structures accordingly. So, the image modifiers establish operations over the images homomorphic to operations over the index structures. However, the operations over the index structures are much less time consuming, so when queries require operations over image constants to be used in the image comparisons, those operations over images can be mapped to operations over the index structures. This leads to a new range of optimizations that can be explored in the image content-based queries, to be explored in future.

6. References

[1] G.F. Dutton, "The Current State of PACS", *Applied Radiology*, Aug. 1990, pp 15-19.

[2] S. Chang and A. Hsu, "Image Information Systems: Where do we go from Here?", *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, No. 5, Oct 1992, pp. 431-442.

[3] A. F. Cardenas at all, "The knowledge-based Object Oriented PICQUERY+ Language", *IEEE Trans. on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 644-656, Aug. 1993.

[4] V.N. Gudivada, V.V. Raghavan, "Content-Based Image Retrieval Systems", *IEEE Computer*, Vol. 28 , No. 9, Sep. 1995, pp 18-22.

[Flickner_95] M. Flickner et alli, "Query by Image and Video Content: The QBIC System", *IEEE Computer*, Vol. 28 , No. 9, Sep. 1995, pp 23-32.

[5] C. Traina Jr. and J.F.W. Slaets, "The Object Representation Model ", (in portuguese) *Technical Report of ICMSC* - University of São Paulo Nº 104, Feb. 1992.

[6] E. Y. Senzako, "SisMatch - Matching for Magnetic Resonance Imaging", (in portuguese) MS Theses, University of São Paulo Oct 1996.

[7] R. R. Santos, A.J.M.Traina and C. Traina Jr., "An Image Definition and Manipulation Language for Object Oriented Data Base" (in portuguese) *Proceedings of XI Brazilian Symposium on Database*, São Carlos - SP, Oct. 1996, pp 127-142.

[8] A. Guttman, "New Features for Relational Database Systems to Support CAD Applications", *PhD Theses*, University of California , Berkeley, Jun. 1984.

[9] L.G. Brown, "A Survey of Image Registration Techniques", *ACM Computing Surveys*, Vol. 24,

No. 4, pp. 325-376, Dec 1992.
[10] C.H. Chow and X.C. Chen, “Moment-Preserving Pattern Matching”, *Pattern Recognition*, Vol. 23, No. 5, Oct 1990, pp. 461474.