Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 749–753
ISBN 978-83-60810-27-9
ISSN 1896-7094

Analyzes of the processing performances of a Multimedia Database Server

Cosmin Stoica Spahiu
University of Craiova,
Craiova, Dolj, Romania
Email: stoica.cosmin@software.ucv.ro

***Abstract*—The paper presents an original dedicated integrated software system for managing and querying alphanumerical information and images from medical domain. The software has a modularized architecture controlled by a multimedia relational database management server. The server is designed to manage database creation, updating and complex querying based on several criteria: simple text-based or content-based image query on color or texture feature, extracted from color and gray-scale image.**

***Keyword*—smultimedia; database server; content based retrieval; insert operations.**

I. Introduction

ONE of the domains where a large quantity of alphanumerical and visual information is acquired daily is the medical one. This information is obtained during the patients' diagnosis and treatment process. Some of the imagistic medical data sources are:

- Electronic medical sheets: these files contain information about patients' name, birth date, medical antecedents, signs, main diagnosis, secondary diagnosis, values of the analysis and treatment.
- Medical images that are stored in digital format or images stocked on different media (X-ray film, paper, etc).
- Digital Imaging and Communications in Medicine (DICOM) files – these standard files are produced by the most part of medical devices (echographs, endoscopes, magnetic resonance imaging devices) which are used in patients diagnosis. A DICOM file contains alphanumerical information (patient name, doctor name, consulting date, diagnosis) and one or several images stored in different formats, compressed or uncompressed.

That is why the problem of storing the medical images collections in digital format along with the associated information (patient name, diagnosis, consulting date and treatment), managing the database and executing efficient queries, it is intensely studied in order to find new and more efficient solutions.

There are only few systems on the market that have already integrated algorithms for image processing and features extraction into the medical diagnosis process. Most of the applications use classic Database Management Systems, like: Microsoft SQL Server, MySQL or Interbase.

The support of the The National University Research Council under Grant CNCSIS IDEI 535 is gratefully acknowledged.

The problem is that almost none of these servers offer support for multimedia data. The users have to implement their own algorithms and methods for images processing.

It is presented in this paper an original solution that integrates both the methods needed to process the images and methods for executeing complex queries, based on the content.

The paper has the following structure: Section 2 presents a short overview of the images format, Section 3 describes the algorithms used for characteristics extraction, in Section 4 it is described the system and made an analysis of the images processing performances and Section 5 presents the conclusions.

II. An original Implementation of a Multimedia Database Server

In order to manage content based retrieval for images collections there have been implemented a series of applications. Most of them are using classic Database Management Systems, like: Microsoft SQL Server, MySQL or Interbase.

The main drawback of these systems is that, most of them offer no support for multimedia data (neither for processing, nor for searching).

In [5][6] it is presented an original solution for managing visual information from images. This solution implemented in Visual C++ is a multimedia database management system (MMDBMS) that includes algorithms for extracting texture and color characteristics and for executing content-based visual queries.

This server is designed to manage medium sized personal digital collections having at most few tens of thousands of records. Over this number of records it should be redesigned the indexes system in order to enhance the execution time.

The implementation includes both a server and a client application that can be executed from any internet browser.

An element of novelty for this implementation is that the client application has the possibility to build visual content-based queries directly from the interface.

The elements of this window which permit content based retrieval are (figure 2):

- Similar With – opens the window for choosing the query image
- Select – permits to choose the field (or fields) that will be presented in the results of the query
- From – it represents the tables in database, that will be used for the query

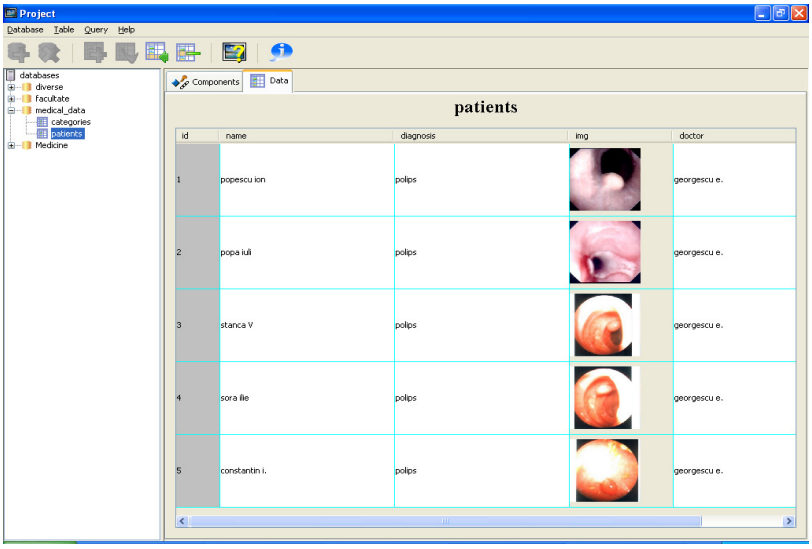

Fig. 1. Main page of the client application

– Where – the image type column used for content-based image query
– Features – it is chosen the characteristic used for content based visual query – color, texture or a combination of them
– Threshold – it is chosen a threshold of accepted similitude between query image and target image. An image with a similitude under that threshold will not be added

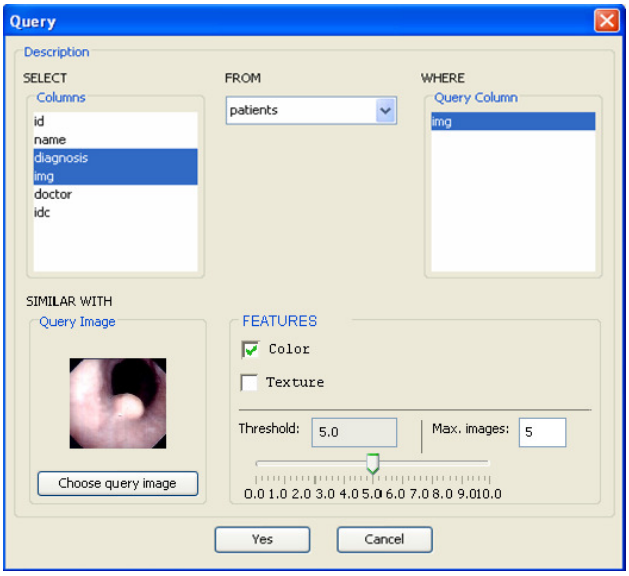

Fig. 2. The window implementing content-based queries

into the resulted query images
– Maximum images – specify the maximum number of images returned by the query

Based on these options, it is generated a modified SQL command for content-based retrieval that is sent to the server along with the query image.

The client communicates with the server through messages exchange using sockets. The client sends SQL commands and receives the results. The results can be either strings (responses to queries or error messages), or multimedia files in which case it is needed to return images to client.

The execution time for content based queries is influenced by two factors: the time needed to process the query image and the time needed to search similar images into the database.

Each image is processed before being stored into the database and extracted color and texture characteristics. That is why it is needed in this step to process only the query image. Depending to the size of the image, this can take an important time.

Only after this step is finished the server can apply specific algorithms in order to find the most similar images.

An original aspect of the server is that it includes all the methods needed to extract the characteristics and compare the similitude of the images.

III. Characteristics Representation and Extraction

The server is intended to be used especially with medical images. The studies have shown that not all the methods used to extract characteristics give the best results in any circumstances [10][11]. That is why it is important to know the domain where the server will be mainly used.

In our experiments there were considered images from gastric tract. For these images the best results were obtained using histograms for color characteristics and Gabor filters for texture characteristics [10][11].

A. Color characteristics representation

Image processing is a technique used to increase the images quality in order to be easily understood by the human eye, or to help extract some important characteristics. In order to succeed it is necessary to identify pixels groups that are interconnected by common characteristics. Some useful information can be extracted from colors repartition in pixels, obtaining the color histogram. For that it is measured and represented the number of pixels for each color. The histogram does not say anything about the geometric relations between pixels. The image-histogram relation is not mutual,

meaning that there can be several images with the same histogram [3].

If the measured characteristics can have k_1 values for the first coordinate, k_2 values for the second…, and k_n for the last, then the resulted histogram will have the size:

$$size = \prod_{i=1}^{n} k_n \tag{4}$$

A major disadvantage of the histograms is that they need a lot of space, especially if the number of colors is high. A good solution would be to store only characteristics that have a non-zero value, or the value is above a specified threshold.

The histograms are invariant to translations, rotations, and they have only small variations when the viewing angle is changed.

The following definition can be given: a histogram represents the colors distribution in an image, region or object. It is calculated using the following formula:

$$h_c[m] = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \begin{Bmatrix} 1, & \text{if } : Q_c(T_c I[x,y]) = m \\ 0, & otherwise \end{Bmatrix} \tag{5}$$

Another disadvantage of the histograms is they do not specify the geometric relations between pixels. That means several images might have the same histogram, although they are totally different.

B. Texture characteristics representation

The texture is the second important characteristic of the images that can be used in content-based retrieval. It is hard to give an exact definition of texture. Usually the word "texture" is used to define the touch feeling of objects, without touching them.

The dictionary definition is: the internal organization of an object, the association modality of an object, characterized by its shape, size of each element and by the geometric relations between each component.

A texture is created by regularly repeating an element or a model on a surface. This model/element is called texture point (textel). The computer graphic techniques define two types of textures: deterministic (regular) and statistic (random).

The deterministic textures are created by repeating the same specified geometric form (e.g.: circle or rectangle). An example of such a deterministic texture is the bricks in the wall.

The statistical textures are obtained by modifying the patterns using specified statistic properties (e.g.: the texture of wood or rocks). They are typically specified by properties of the spatial frequencies.

In order to extract the texture characteristics there have been studied a series of methods. The most representative are Gabor filters and Coocurence matrix. Although there are a lot of other available techniques, there is no one to be considered the best. This depends especially by the type of images used (from nature, medical, etc.).

C. Extracting texture characteristics using Gabor filters

Starting from HSV color space representation, the color can be represented in complex. Any point from the HSV cone can be computed using the formula [4]:

$$z_M = S(\cos H + i \sin H) \tag{6}$$

The saturation can be interpreted as size, the hue as the phase of the complex value. There are a lot of advantages of such a representation. First, it is simple due to the fact that color is a scalar and not a vector and the combination between channels is made before filtering. In conclusion, the color can be represented in complex using:

$$b(x,y) = S(x,y) \times e^{iH(x,y)} \tag{7}$$

To compute the Gabor characteristics for an image represented in HS-complex it is used a method similar as computing the monochromatic Gabor characteristics. This is due to the fact that the color channels are combined before filtering [4]:

$$C_{f\phi} = (\sum (FFT^{-1}\left| P(u,v) \times M_{f,\phi}(u,v) \right|))^2 \tag{8}$$

The Gabor characteristics are created using the value $C_{f,\varphi}$ computed for 3 scales and 4 directions:

$$f = (C_{0,0}, C_{0,1}, \ldots\ldots, C_{2,3}) \tag{9}$$

The similitude between these characteristics is defined by the metric:

$$D^2(Q,T) = \sum \sum d_{f,\phi}(Q,T), where\ d_{f,\phi} = (f^Q - f^T)^2 \tag{10}$$

IV. Experiments and Results

Because it is a client-server application, the execution time depends on two aspects: the network speed that is used to send data to the network and the data processing speed.

Table 1.
Time needed to transfer images

No	Image resolution	Image size	Time needed
1	160 ×160	75 KB	0,30 s
2	240 × 240	172 KB	0,75 s
3	480 × 480	675 KB	2,90 s
4	640 × 640	1204 KB	5,20 s
5	1200 × 1200	4220 KB	18,30 s

Usually the information sent to server is based on strings, less than 100KB in size. Nowadays the network speed is high enough that the sending time can be considered close to zero.

In our case, an important role in data transfer is represented by the images data. Depending to the resolution and compression used, they might have a size up to several MB. In this case it is very important the speed used to send data to the network.

Taking into account that the size of a BMP image with resolution of 100×100 pixels, is 30 KB it can be deducted the time needed to send images to the server:

The time effectively needed by the server for an insert operation is higher than presented above because the server uses a handshaking communication technique for client communication and data sending: the client sends to the server the insert command, the server responds with an OK confirmation. It analysis the data next and if needs more data (such as receiving an image), sends to the client a data request. The client responds sending the image. To the last step, the server acknowledges receiving the image. If one of them does not receive an acknowledgement it will resend the information.

After all the information is receive, before making the insertion in the database, the server has to process the image received and extract the color and texture information.

In order to extract the characteristics, the image colors have to be translated from RGB to HSV color space, quantized to 166 colors and after that applied the algorithms presented above to extract histogram and texture. These operations are made to the pixel level and need a high processing power and important execution time.

In the tested version, the server process only BMP images. If the image would be compressed (e.g.: JPEG), it should be decompressed first and only after that processed as the BMP images.

In Table 2 it is presented the time needed to extract the color and texture characteristics from images having different resolutions.

The graphic resulted from the above table is presented in figure 2.

It can be noticed a linear growth of the execution time with the image size, due to the linearity of the algorithms used.

TABLE 2.
TIME NEEDED TO PROCESS THE IMAGES

No	Image resolution	Time for color characteristics	Time for texture characteristics	Total time
1	160 ×160	0,45 s	14 s	14,45 s
2	240 × 240	0,60 s	14,5 s	15,10 s
3	320 × 320	0,70 s	15,30 s	16 s
4	480 × 480	1 s	63,70 s	64,70 s
5	640 × 640	1,50 s	108,50 s	110,50 s

For an image having the 160 × 160 pixels resolution it is necessary approximate 15 seconds to extract the characteristics. For an image of 640 × 640, the time will increase to almost 2 minutes.

Taking into account that there are cases when several users want to execute insert operations simultaneously, it was considered to be useful to limit the images resolution to 500×500 pixels. This will give the possibility to finalize the insert operations in a reasonable time.

V. CONCLUSIONS

The paper presented a new solution for managing and querying multimedia images collections. The implemented

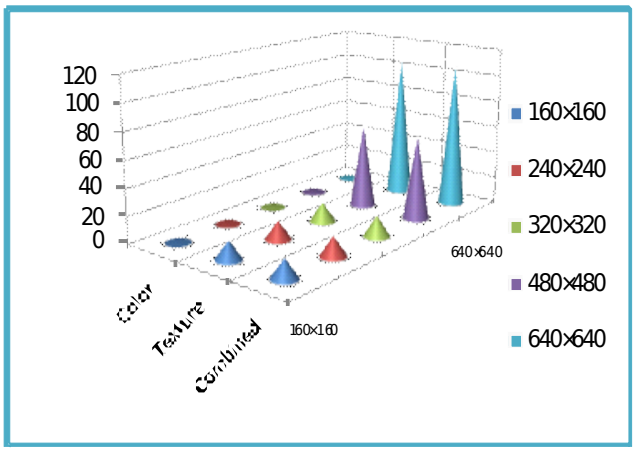

Fig. 3. Time vs. Image resolution

multimedia relational database management system includes an original data type, called IMAGE used to store all the characteristics extracted from images. These characteristics are used in content based retrieval.

It is created for managing and querying medium sized personal digital collections that contain both alphanumerical information and digital images (for examples the ones used in private medical consulting rooms). The software tool allows creating and deleting databases, creating and deleting tables in databases, updating data in tables and querying. The user can use several types of data as integer, char, double or image.

The quality of the server is tested from the execution point of view. It is tested the time needed to process the insert queries, namely the time needed to extract color and texture characteristics.

This software can be extended in the following directions:

- Adding new types of traditional and multimedia data types (for example video type or DICOM type - because the main area where this multimedia DBMS is used it is the medical domain and the DICOM type of data is used for storing alphanumerical information and images existing in a standard DICOM file provided by a medical device)
- Studying and implementing indexing algorithms for data inserted in the tables in order to enhance the execution time.

REFERENCES

[1] D. Cardani, *"Adventures in HSV Space"*
[2] D. Cojocaru, *"Images Acquisition, processing and recognition"*. Universitaria Craiova, 2005
[3] C. Palm, D. Keysers, T. Lehmann, K. Spitzer, "Gabor Filtering of Complex Hue/Saturation Images For Color Texture Classification". In *5th Joint Conference on Onformation Science (JCIS2000)*, pp. 45-49, 2000
[4] C. Stoica Spahiu, C. Mihaescu, L. Stanescu, D.D. Burdescu, M. Brezovan, "Database Kernel for Image Retrieval". In *The First International Conference on Advances in Multimedia (MMEDIA 2009)*, Colmar – France, pp. 169-173, 2009
[5] C. Stoica Spahiu, L. Stanescu, D.D. Burdescu, M. Brezovan, "File Storage for a Multimedia Database Server for Image Retrieval". In *The Fourth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2009)*, Cannes/ La Bocca, France, pp.35-40 (2009)

[6] M. Kratochvil, *"The Move to Store Images In the Database",* 2005 http://www.oracle.com/technology/products/intermedia/pdf/why_images_in_database.pdf

[7] G. Lu, *"Multimedia Database Management Systems"*. Artech House Publishers, 1999

[8] H. Muller, A. Rosset, A.Garcia, J.P.Vallee, A. Geissbuhler, *"Benefits of Content-based Visual Data Access in Radiology"*. Radio Graphics. 25, pp. 849-858, 2005

[9] L. Stanescu, D.D. Burdescu, M. Brezovan, *"Multimedia Medical Databases Chapter Book"*. In: Sidhu, Amandeep S.; Dillon, Tharam; Bellgard, Matthew (Editors.), Biomedical Data and Applications, Series: Studies in Computational Intelligence, Vol. 224, Springer Verlag, 2009

[10] L. Stanescu, D. D. Burdescu, C. Stoica Spahiu, M.Brezovan, "A study of two color systems used in content-based image query on medical imagery". In *International Conference on Informatics in Control, Automation and Robotics* (ICINCO 2007). pp. 337-340, 2007

[11] T. Gevers, A. Smeulders, Color-based object recognition. *Pattern Recognition*, Vol. 32, pp. 453-464, 1999.

[12] HSV Color Space, http://en.wikipedia.org/wiki/HSV_color_space.