International Journal of Computer Science and Applications,

Vol. 7 No. 3, pp. 140 - 155, 2010

A MULTIMEDIA DATABASE SERVER: IMPLEMENTATION AND FUNCTIONS

COSMIN STOICA SPAHIU
University of Craiova, Craiova
stoica.cosmin@software.ucv.ro

The paper presents the implementation and performances of a Multimedia Database Server software tool used for managing medium sized image collections. The software tool uses traditional data types (integer, double, char, varchar) and a new complex data type, called image. This type is used for storing the image along with the information extracted. This refers to the image's size and type, color and texture characteristics. An element of originality for this software is the content based retrieval module that allows the user to build content based visual queries to the image level. The server's performances are studied from the quality of the retrieval point of view. For the tests there are used both color and texture characteristics which are automatically extracted.

Keywords: multimedia database server; image data type; content based retrieval.

1. Introduction

With the development of the different technologies, in the last years an impressive number of gray and color images have been accumulated. The development of the multimedia field, the creation of new images and video archives of large dimensions, have led a series of researchers to turn their attention, over the past decade, towards creating new tools for retrieving the visual data based on their content.

The purpose of this process is to retrieve from a database the relevant images for a query. It represents in fact, an extension of the traditional process of information retrieval to visual media. The term "content" refers to specific characteristics of the images that can be automatically extracted. The most common characteristics used for content based retrieval are: texture, color and shape [Del Bimbo (2001)][Faloustos (2005)][Gevers (2001)][Gevers, Smeulders (1999)][Burdescu et al. (2009)].

This paper presents an application that includes a dedicated MMDBMS server based on the SQL3 standard, which is less expensive than a commercial database server. The system is platform independent and can easily manage medium sized image collections and alphanumerical information from the medical domain. It has a standard client interface for building content based retrieval using color and texture characteristics and can be easily used by any person working in this area, even if he does not have advanced knowledge in using the computer.

This MMDBMS has the following original elements and advantages:

- The way that images are managed (a new data type used to store images and characteristics)

- Integrated methods for extracting characteristics and executing content based queries
- Client visual interface for content-based image query using color and texture characteristics
- Possibility to see the images when the records are viewed

The paper has the following structure: section 2 presents an overview of similar implementation of well known database servers. Section 3 presents the implemented application, section 4 presents the main functions of the MMDBMS, in the section 5 there are presented the execution experiments and section 6 presents the conclusions and future work in this project.

2. Similar Implementations

In order to manage content based retrieval operations for images collections, there have been implemented a series of applications. Most of them they use classical database systems such as: Microsoft SQL Server, My SQL or Interbase.

We will try next to exemplify few of the most representative database servers, from the functionality point of view (possibility to store multimedia data) and from the data storage algorithms point of view.

2.1. *Adaptive Server Anywhere/Advantage Server Anywhere*

The first database servers analyzed are the one produced by Sybase Company, namely Adaptive Server Anywhere (ASA) and Advantage Server Anywhere (ADS).

These servers contained at the beginning the same module as Microsoft SQL Server. Up to the version 4.9, these products were almost identical [ASA (2008)][Jensen, Anderson].

Advantage Database Server is a relational database server with good performances, which supports SQL commands and data indexing. It has the advantage of having a low cost comparing to other products and it is easy to be installed.

Its basic characteristics include data encoding on disk, using partitions and a new technology for queries processing that is currently on the patenting process.

In order to manage unstructured data, ASA/ADS created two kinds of data types that are similarly processed. The first data type is CS_TEXT for text information and the second is CS_IMAGE for image data information [Jensen, Anderson][Sybase (a)][Sybase (b)].

CS_TEXT can contain data variable in size, not greater than 2147483647 bytes.

CS_IMAGE is used for image data type information. It describes a variable length attribute, with a size up to 2147483647 bytes, stored binary. CS_IMAGE is defined as unsigned char data: typedef unsigned char CS_IMAGE.

Because the size of this kind of attributes can be very high, the values are not stored inside the table. It will be stored only a pointer of text data type to a special file used only to store this type of information. To avoid the possibility that different data with the same content are not overlapping, the image type column will have associated a time stamp.

To work with these data types, the server offers few basic functions to load the images, transfer the images and get images size.

To send this data, the clients send information as an undifferentiated data flux using a special function write_text_stream. Because it is not different in parameters, the server cannot use methods frequently used to compute input data: srv_descfmt, srv_bind, and srv_xferdata. It will have to use a set of methods specialized for text and images.

To send text or image data back to the client there are two possibilities depending to the number of columns from a record. In case the record has only one column and that column has the image or text type it can be considered as an undifferentiated data flux. If the returned record contains more columns, including a text/image column, then it will be processed using describe/bind/transfer.

When it is added an information of image or data type, it will be reserved 16 bytes space inside the text/image column, for a pointer. In this case a part of it represents page number of the text/image file header. It is known with the name of first text/image page (FTP).

FTP contains two parts:

- Text Data Page link that contains the useful information
- Optionally, the data node structure used to access the information in an easy manner.

Once this FTP is allocated for text/image data, it will never be de-allocated. If a future update of data structure needs a smaller number of pages that were initially allocated, the pages no longer in use will be de-allocated. If all the pages will be set the value to NULL, all pages will be de-allocated excepting FTP.

The figure 1 presents the relation between records containing images/text and the text/images pages.

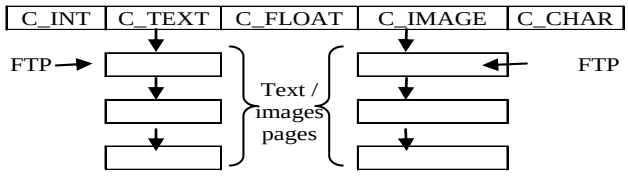


Fig.1. Structure of a table that contains text and image type information.

ASA/ADS consider text and image data as being identical, with only one exception, namely the character set used for conversion.

2.2. *My SQL*

MySQL server does not offer any other type that can be used to store images, excepting BLOB data type. BLOB data is an object stored in binary form. It can be used to store a variable amount of data. The BLOB data attributes doesn't have a predefined character data set. They are computed as binary strings (arrays of bytes). Data sorting and comparing can be done only using numeric values of the attributes – numeric values on the bytes from records [MySQL (2009)].

There are four types of BLOB data: TINYBLOB, BLOB, MEDIUMBLOB and LONGBLOB. The only difference between them is the maximum size accepted for data.

If the SQL strict mode is not active and a value that exceeds the maximum size is put in a BLOB column, it will be truncated to the maximum size. If the strict SQL is used, when a data bigger than the maximum size will be inserted, it will not be inserted and it will be returned an error message.

The BLOB data type can be considered to be similar with VARBINARY. This can have an unlimited size. The difference is that if you try to index a BLOB data type it has to be specified the desired size of the index. Another difference is that a BLOB data type cannot have default values.

2.3. *Microsoft SQL Server*

Microsoft SQL Server treats the image and large text type different from the rest of available data types. The most important aspect that should be considered is that images need a large amount of disc space. That is why it is recommended to store large data (having the size bigger than 8KB) in external files, outside MSSQL Server. In this case it will be stored on the server only the path to the file that has the information. This kind of implementation gives the user the possibility to store large amount of data on different disks and even on a different server.

The big advantage of this approach is that the database can be kept to a low size. It should be mentioned that the default initial size of the database is 20MB, and the future extensions will be made in 5 MB steps.

A possible disadvantage of this approach is that the users have to be careful do not delete the image files from the computer before deleting them from the database.

If the solution to store the images inside the tables is chosen, a first option would be to use VARBINARY(max) data type. This can store a date having up to 231 bytes, namely 2GB [MS SQL Server]. A big disadvantage is that SQL Server stores a history of the data before and after any update that generating a high volume of data needed to be stored on the disc. This history is used in case of an error, to restore previous data.

It should be monitorized the available disc space to avoid situations when the server suddenly stops and doesn't execute any query because of the lack of space on the disk.

Microsoft SQL Server also offers to the users an "image" data type, but with no other support. It is considered a binary data with a variable length, between 0 and 231-1 (2,147,483,647) bytes. There are no predefined functions that can be used to extract images characteristics or building content based queries.

More than that, in the SQL Server 2008 it is specified the fact that the "ntext", "text" and "image" data types will be eliminated in a future version of the server. The recommendation was to avoid using these data types in the applications that are currently implemented and to try to change the already implemented applications in order not to use them. Instead of them there should be used nvarchar(max), varchar(max), or varbinary(max) [Dewson (2006)][MS SQL Server][Chigrik (2007)][SQL Server (2008)].

For the MS SQL Server, the images cannot be seen directly from server. It is needed an external program that can do that.

2.4. *Oracle 10g*

The full solution is provided by Oracle – Oracle 10g Database Server and Intermedia tool that support all kind of multimedia data types, including DICOM files.

It offers support for the images by ORDImage data type. In the Oracle 10g Intermedia version it is also offered support for the ISO/IEC standard 13249-5:2001 SQL/MM Part 5: Still Image Standard. On this standard there are defined relational data types for images and images' characteristics. Each object type includes attributes, methods and it has associated SQL functions and methods. By using the SQL standard as an interface, gives the possibility for the applications to be more accessible and easy to be transferred on different database servers. The following object types are defined by SQL/MM Still Image:

- SI_StillImage – stores digital images along with the asociated characteristics (height, wide, type, etc)
- SI_AverageColor – describes the medium color characteristics of an image.
- SI_Color – has the color values of a digital image.
- SI_ColorHistogram – describes the relative frequencies of the colors inside the image.
- SI_PositionalColor – describes the color characteristics of an image from the location point of view.
- SI_Texture – describes the texture characteristics of an image.
- SI_FeatureList – describe an image that is represented by composed characteristics, based on four basic characteristics along with associated values [OracleDatabases] [Oracle (2005)].

This type of solution is not always justified and cannot be always implemented by small companies. The reason is that it implies high costs to buy the server and to design complex applications for content based retrieval.

A multimedia database server is composed by two components: an application instance and a system that manages information on the disc [OracleDatabases][Oracle (2005)][Chigrik (2007)].

The Oracle database memory architecture on the server side is called SGA (System Global Area). SGA keeps information in the cache about: data buffers, SQL commands, and users' information. The server also keeps log files with the transactions history.

Oracle keeps information in a logic form as tables and physical as files. Tables can contain different types of memory segments as data segments, indexes segments, etc. In return, the segments contain one or several sub-segments. These sub-segments are contiguous groups of data blocks. A data block is the basic unit for storing information. To the physic level, the data files contain one or several data blocks, where the size of blocks can be different from one file to another.

The Oracle server manages its storing capacity using the information kept in the SYSTEM tables. These contain data dictionary and indexes and clusters. A data dictionary is a collection of tables that contain all the information about the objects declared by the users in the databases.

Each Oracle instance reserves a memory buffer when it starts and it is released when it is stopped. The following information are stored in the SGA, each of them having the size defined when the server starts:

- Buffer cache: keeps the most recent data blocks that were not written yet on the disc.
- REDO log file: has a log with all the changes that were made in the database. It is used in case on an error.
- Common area: this area keeps common structures and data from data dictionary

The data dictionary from the cache includes a series of tables and views that forms a map with the database structure. Oracle stores information both about physical structures of the database and logical ones.

The Global Program Zone (PGA) contains control data for Oracle server processes. The size of the PGA depends on the options set up when installing. These options refer to:

- Stack size: the memory that stores the variables from current session.
- Session information: information about the status of current information
- SQL zone: needed variables are stored in the buffer
- Sorting zone: a zone where there are stored information about sorting, indexes, etc.

3. The MMDBMS Implemented for Images Management

It can be observed that excepting the Oracle Database server, the other database servers analyzed above does not offer full support for images management. Most of them recommend using the BLOB data type, with no other support. Nor Oracle is always the best solution, because of the high cost involved, that making hard accessible for small companies.

As an alternative it has been implemented an original Multimedia Database Management Server (MMDBMS) that has included a new data type called IMAGE. This data type can be used to store images along with color and texture characteristics automatically extracted from images [Stoica Spahiu (2009)][Stoica Spahiu et al. (2009)].

The server is a tool implemented in C++ that has integrated special functions for creation, maintenance and simple text based query. Other original elements of the server are the special functions used for color and texture characteristics extraction and for executing content-based visual query. These methods are also integrated into the server.

The tests that have been performed were done on images from medical domain where changes in color or texture are very important. Not all the methods generate good results in all the situations. Depending to the type of images taken into account, there should be chosen the right methods. For medical images, the studies have shown that the best results were obtained using color histograms for color characteristics and Gabor filters for texture characteristics [Gevers T., Smeulders, W.M. (1999)][Stanescu et al. (2007)].

This MMDBMS permits database creation, table and constraints adding (primary key, foreign keys), inserting images and alphanumerical information, simple text based query and content based query using color and texture characteristics. The software tool is easy to be used because it is according to the SQL standard. It does not need advanced informatics knowledge and has the advantage of low cost. It is a good alternative for a classical database management system such as MS Access, MS SQL Server, Oracle10g Server and Intermedia, which would need higher costs for database server and for designing applications for content-based retrieval.

The server is implemented in a modular way, being easy to add new functionalities or to update the already existing modules. It has two main modules: server engine and server files manager.

The server engine includes all functions implemented in the DBMS. It is composed from several sub-modules each of them with specific tasks: the main module, Queries Response module, Select images processing module, Characteristics extraction module, Update/Delete module.

The main module is the module which manages all communications with the client. It is the one that receives all queries requests, check what is the type of query requested, extracts the parameters of the query and calls the specific module to execute it. All the actions of the server are managed by it.

Queries response module is used after the server receives a query command to be executed. The results will be sent to it in order to be compacted using a standard format. Then return the results o the client. The client will receive it on the same communication channel used to send the request.

Select/Select/Image Processing module. If the main module concludes that is a SELECT SQL command, it will call the Select Processing module. This module extracts the parameters from the query and then search in the database files for specific information. If the query is a SELECT IMAGE query, it will use for comparing the similitude of characteristics instead equality of parameters.

Characteristics Extraction Module is used when the main module receives a SELECT IMAGE or a UPDATE query which uses an image that is not already in the database. In this case it is needed first to be processed. This module is called to extract the color and texture characteristics of the image. The data of the results will be used to initialize a variable of IMAGE data type.

Update Processing Module. When the query received from the user is an UPDATE command, this module will be called to execute it.

Delete Processing Module it is called when the user executes a DELETE command. The kernel executes only logic delete operations. It never executes physical deletes. The physical deletes are executed only when a “Compact Database” command is sent by the user.

The second main module is the Database Files Manager. It is the only module that has access to files. It is the module that manages the users’ access to database adding or deleting locks to the files. When a client module request a read form a file it is enabled a read lock for the specific file (that represents a table in the database). All other read

requests will be permitted but no write requests will be allowed. If the client module request a write to file, it will be enabled a write lock. No other requests will be allowed until the lock is canceled.

In the next figure is presented the architecture of the implemented multimedia database server:

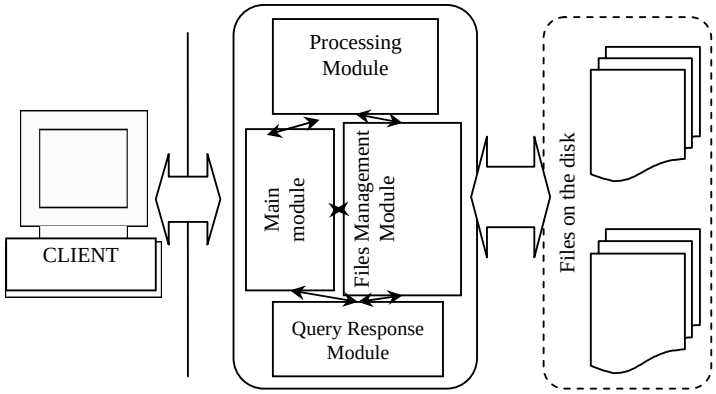


Fig. 2 Architecture of the MMDBMS

4. The Main Functions Implemented for the MMDBMS

The functions provided by this MMDBMS are [Stoica Spahiu (2009)][Stoica Spahiu et al. (2009)]:

1. Managing databases structures and information

The functions refer to the following operations: browsing databases, create/delete databases, create/delete tables, defining constraints for the tables (primary keys referential integrity), update information in the tables (insert, update, and delete records).

The most important commands recognized by the system, are:

(1) create database <database_name>

eg: create database student

This command creates a new database on the server.

(2) use database <database_name>

eg: use database student

Before being able to execute operations on it, first it should be defined the default database.

(3) create table <table_name> (<atribute1> <data_type1>, <atribute2> <data_type2>, [<atribute3> <data_type3>,…])

The command will create a new table in the defined database with the specified attributes. A table can have any number of attributes. The data types that can be used are:

- integer
- double
- varchar

- image

If it is used the varchar data type, it is needed to be specified the maximum size of the array.

(4) alter table <table_name> add primary key (<attribute_name>)

eg: alter table person add primary key (id)

The command will define <attribute_name> as primary key in <table_name>. Using this command for several times, there can be defined composed primary keys.

(5) alter table <modified_table_name> add foreign key (<atribute1_name>) references <table_name2> (<referred_ atribute2_name>)

eg: alter table person add foreign key (id) references picture(id)

The command defines a foreign key between two tables <modified_table_name> and <table_name2>. The key will link the two attributes <atribute1_name > and <referred_ atribute2_name>.

Using this command for several times, there can be defined composed foreign keys. It cannot be defined the many-to-many type links. That is why at least one of the attributes used must be a primary key.

(6) get table keys <table_name>

eg: get table keys person

1. insert into table_name values (value1, value2, [value3, …])
2. insert into person values(1,'George O.',20.5)
3. insert into imagine values(1,'George O.',' analysis.bmp')

In the table <table_name> there will be added the values between parentheses. It is mandatory to add values for each attribute. There should not be any NULL attributes. The order of the inserted values should be the same as the one of the attributes in the table

The text based query can be used on any attribute defined in the table. It is used the SQL syntax, having the possibility to specify to return only certain attributes from a record, or all the attributes. There are accepted any number of conditions specified in the WHERE clause, combined by AND or OR operators.

The syntax is:

(7) select * from table_name where attribute1 [</=/>] values [and attribute2 [</=/>] values2 …]

eg: select * from person

select * from person where age>45.5 and id<40 and name= 'Adrian Ionescu'

An element of originality is that when seeing the database records, the images can be automatically displayed.

When inserting an image in an Image data record, after the image is selected, the module for processing images is activated. It will automatically extract color information represented by the color histogram with 166 colors and the texture information with 18 values.

The DBMS makes a series of validations both in the design phase and after it, when populating database: unique names for databases, tables, columns, adding data, etc.

2. Content-based visual query

The MMDBMS offers the possibility to build the content based visual query, in an easy manner, at the image level. The elements that can be used to build the content based

retrieval query are: similar with, select, from, where, features, threshold and maximum images returned.

The syntax of the query is:

SelectImage [/<attribute_name1>, <attribute_name2>, …] FROM <table_name> WHERE <image_column_name> LIKE QueryImage (METHOD: color[,texture] MaxImages <no_of_returned_images>)*

Where:

"attribute_name1" is the attributes that should be returned

"table_name" is the name of the table

"image_column_name" is thename of the column that contains the image itself

"no_of_returned_images" specifies the maximum number of images returned

Eg:

SelectImage name, picture from picture where picture Like QueryImage (method: color maxImages 5)

selectImage * from Patients where age>50 and picture like queryimage

The retrieval operation uses two functions to calculate dissimilitude between images: one for the texture (Euclidian distance) and one for the color characteristics (the histogram intersection). The user has the possibility to choose one or both of them. If both distances are used for the query, the total distance is the arithmetical average between them.

The system creates in the background a modified SQL Select command, adapted for content-based image query, using the user preferences. It has the following structure:

Select doctor.name, doctor.img From personnel where doctor.img Similar with Query Image (method : color, max.images 5)

This modified Select command specifies that the results are obtained from personnel table. The values from names field and the images similar with the query image taking into consideration the color feature are displayed. The result will contain only first 5 images. In the resulting set, it is also presented the similitude distance between the query and the target images.

3. Administrative functions

The MMDBMS offer support for users' management (creating, deleting users and setting their rights), retrieving databases structure and compacting databases after successive delete/insert operations. For each user there are two kind of rights that can be set up: general rights (creating other users or other databases) and particular rights (rights for a specified database), given by the administrator or by the database owner.

5. Server Performances

The main goal of the indexing operations in a database and of choosing the best algorithms for computing the images similitude, is to obtain a better execution time and a high efficiency. The system performances can be measured by the execution time and by the quality of the retrieved information. In the case of visual retrieval, the quality of retrieval was calculated using two parameters:

- recall: $$r = \frac{no.of\ relevant\ \ images\ \ \cap no\ of\ retrieve\ d\ images}{total\ \ no\ of\ relevant\ \ images} \quad (1)$$
- precision: $$p = \frac{no.of\ relevant\ \ images\ \ \cap no\ of\ retrieved\ \ images}{total\ \ no\ of\ retrieved\ \ images} \quad (2)$$

For the experiments it was created a database with 170 images, with the following diagnosis:

- polyps - 75 images inserted into database
- esofagitis - 30 images inserted into database
- rectocolite - 29 images inserted into database
- tumor - 36 images inserted into database.

The images that were used for tests had the diagnosis “esofagitis”.

The quality of the retrieval was tested using both color and texture characteristics. There were performed several tests specifying that the number of similar images returned by the server to be: 5, 9 and 15.

For each case it was selected the same query image presented in figure 3.

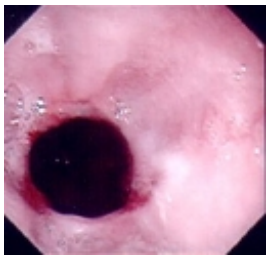
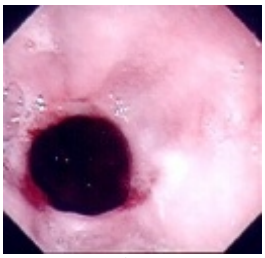


Fig. 3 The query image used for tests

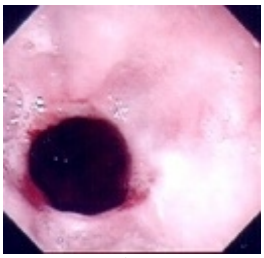
5.1. *Test 1 – 5 images returned*

First it was that the server should return only 5 images, the most similar. The images are returned in descending order of the similarity.

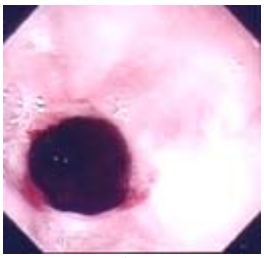
The results are presented in figure 4.



esofagitis



esofagitis



esofagitis


esofagitis

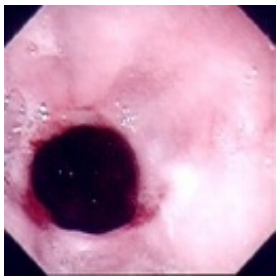

esofagitis

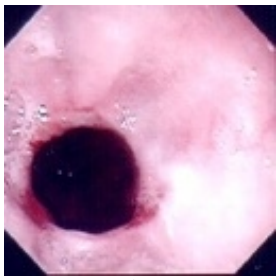
Fig 4. The five images returned by the system using color and texture for computing the similitude

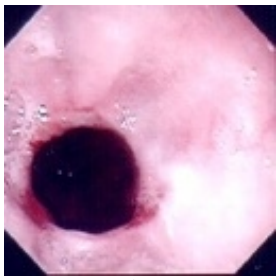
For these results there were calculated the precision and recall parameters:

$$p = \frac{5}{5} = 1 \qquad r = \frac{5}{30} = 0{,}16 \tag{3}$$

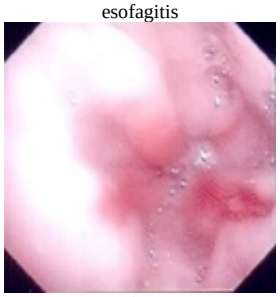
5.2. *Test 2 – 9 images returned*


esofagitis


esofagitis


esofagitis


esofagitis


esofagitis

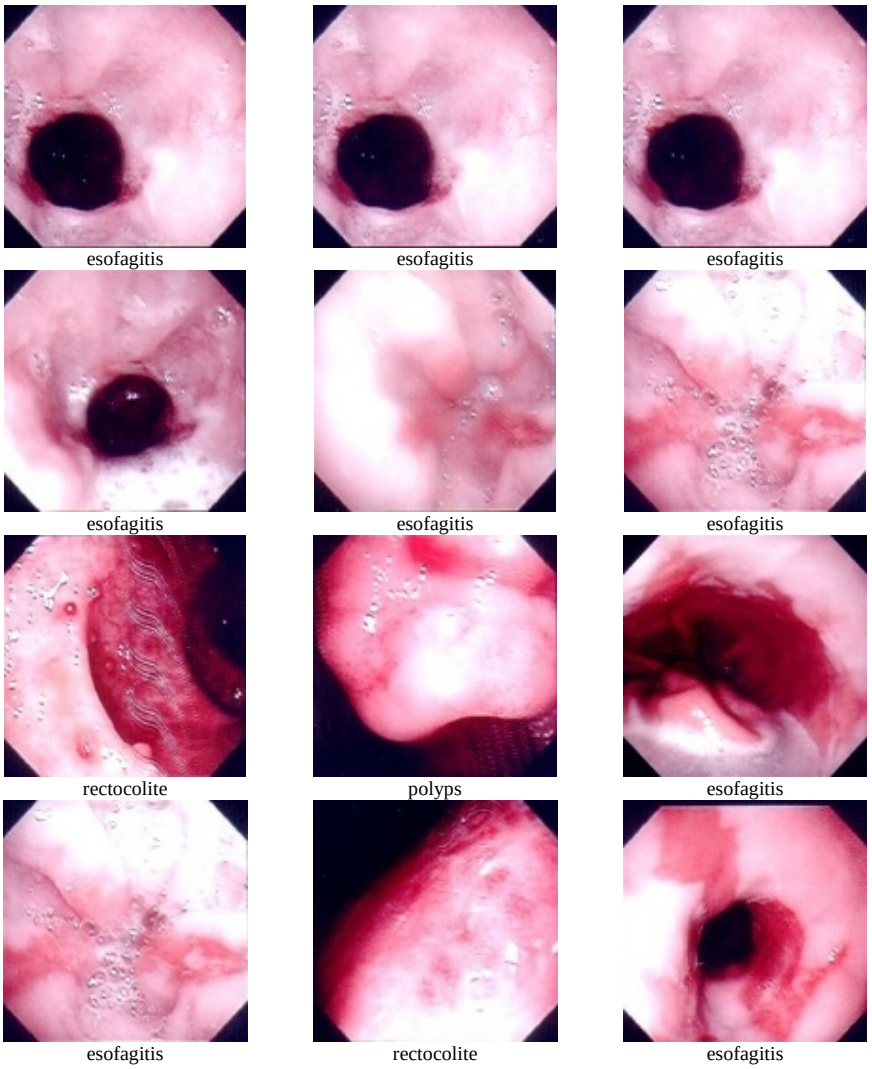

esofagitis

Fig 5. The nine images returned by the system using color and texture for computing the similitude

For these results there were also calculated the precision and recall parameters:

$$p = \frac{7}{9} = 0{,}77 \qquad r = \frac{7}{30} = 0{,}23 \tag{4}$$

5.3. *Test 3 – 15 images returned*



esofagitis esofagitis esofagitis

Fig 6. The fifteen images returned by the system using color and texture for computing the similitude

The calculated precision and recall parameters are:

$$p = \frac{12}{15} = 0{,}8 \qquad r = \frac{12}{30} = 0{,}40 \tag{5}$$

Based on the results obtained in test 1, 2 and 3 it was obtained the following graphic:

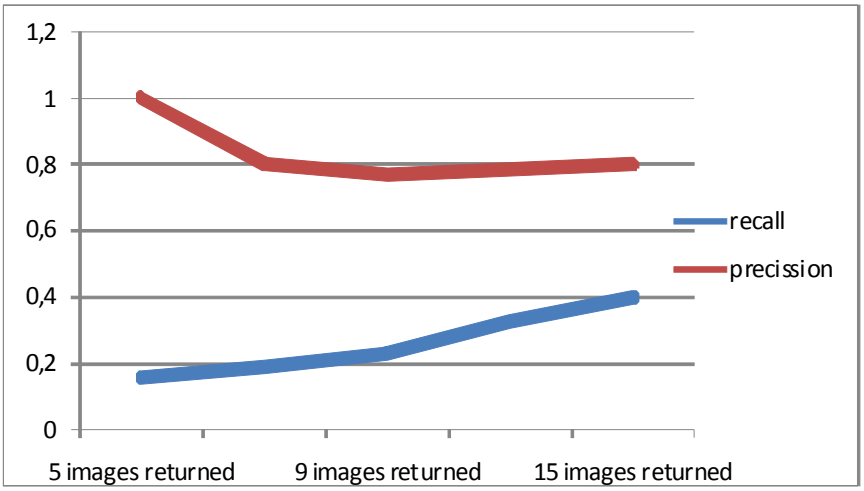


Fig 7. Precision vs. recall graphic

The test concluded that the results were good. A high number of relevant images were retrieved from the total number of images returned.

Dependind to the users' preference, the system can either enhance the returned images' relevance by lowering the number of returned images (the precission parameter will have higher values), or enhancing the recall parameter value by increasing the number of images returned.

It is also recommanded to use both color and texture characteristics. Depending on the images type there are cases when it gives better results the color characteristic and cases when gives better results the texture characteristic. In this way, the images that are not retrieved by one of the methods, will be retrieved by the second.

6. Conclusions and Future Work

The paper presents the design and functions implemented for a multimedia database management that can be used for managing medium sized collections of images. There is also made a survey of the similar implementations of well known database servers. An element of originality is the use of the image data type that stores both images and image characteristics that were automatically extracted. The methods that were implemented for this server are in compliance with the SQL/MM standard.

When an image is added to database, it is processed and stored along with the extracted characteristics using Image data type. The color information is represented in HSV color space, quantized to 166 colors. For texture the co-occurrence matrix is used.
The function that brings originality to the MMDBMS is the content based visual query using color and texture characteristics. The client has the possibility to execute query in a visually manner or by using a modified Select command, adapted to this type of query that is specific for multimedia databases.

In the last chapter there are analyzed the performances of the server from the retrieval quality point of view. There are made several tests on a database containing 170 images from the medical domain. The results are encouraging as the number of relevant images returned is high.

On the next step there should be focused on the execution time of the retrieval and search for new methods in order to improve the execution time for when searching in large databases.

The application can be enhanced in the future by adding new recognized data type. From the multimedia data point of view, it can be added new methods for processing video and audio data. In this way the content based retrieval can be extended to these new data types.

References

Adaptive Server Anywhere Data Storage White Paper-Technical: Mobile Enterprise, Database Management. 2008. http://www.sybase.com/detail?id=1009483.

Burdescu, D., Brezovan, M., Ganea, E., Stanescu, L. (2009): A New Method for Segmentation of Images Represented in a HSV Color Space, Advanced Concepts for Intelligent Vision Systems (ACIVS'2009)

Chigrik, A. (2007): SQL Server 2000 vs Oracle 9i, http://www.mssqlcity.com/Articles/Compare/sql_server_vs_oracle.htm

Del Bimbo, A. (2001): Visual Information Retrieval, Morgan Kaufmann Publishers. San Francisco USA

Dewson, R. (2006): Beginning SQL Server 2005 for Developers. s.l. : Springer-Verlag New York

Faloutsos, C. (2005): Searching Multimedia Databases by Content. Springer

Gevers, T., Smeulders, W.M. (1999): Color-based object recognition. Pattern Recognition. 32, 453-464

Gevers, T. (2001). Color in Image Search Engines. Principles of Visual Information Retrieval. Springer-Verlag, London

Jensen, C.; Anderson, L. A First Look at Advantage Database Server http://www.sybase.com/files/White_Papers/Sybase_advantage_first_look_ads9_wp.pdf.

Microsoft SQL Server.

http://en.wikipedia.org/wiki/Microsoft_SQL_Server.

MySQL 5.0 Reference Manual. (2009)
http://dev.mysql.com/doc/refman/5.0/en/blob.html.

Oracle Databases.
http://en.wikipedia.org/wiki/Oracle_database.

Oracle (2005), Oracle InterMedia: Managing Multimedia Content
http://www.oracle.com/technology/products/intermedia/pdf/10gr2_collateral/imedia_twp_10gr2.pdf

SYBASE - Open Server Server-Library/C Reference Manual
http://download.sybase.com/pdfdocs/cng1250e/osref.pdf

Sybase. http://en.wikipedia.org/wiki/Sybase.

Stoica Spahiu C. (2009): A Multimedia Database Server for information storage and querying. 2nd International Symposium on Multimedia – Applications and Processing (MMAP'09), Vol. 4, pp. 517 – 522.

Stoica Spahiu C.; Mihaescu C.; Stanescu L.; Burdescu D.D.; Brezovan M.(2009): "Database Kernel for Image Retrieval". The First International Conference on Advances in Multimedia (MMEDIA 2009), pp. 169 – 173

SQL Server 2008 Books Online. (2009): http://msdn.microsoft.com/en-us/library/ms187993.aspx.

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