MMDB:NICE - MULTIMEDIA IMAGE DATABASE WITH IMAGE CONTENT SEARCHING

Muguras D. Mocofan
"Politehnica" University of Timisoara, Electronics and Telecommunications Faculty, Bd. Vasile Parvan, No. 2, 1900, Timisoara, email: mugur@mail.opendrum.utt.ro

Abstract – For the indexing of an image data bases can be used many features. In this paper are used features like histograms for the colour components, stohastic moments, few parameters of the co-occurrences matrix and the Gabor filters decomposition. Using all of them, I obtain good performances in the field of content-based image indexing and retrieval. There are some speed problems because there are many features used in the search process. Here, I present an algorithm for indexing and retrieval, which uses a tree algorithm for speed increasing.

The area of applications is very wide: multimedia documents, transaction systems, medical application, query and browsing.

Key words: multimedia databases, image content searching.

INTRODUCTION

This paper presents an interactive multimedia database application that combines the advantages of object-oriented programming and the image content search. The model proposes a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases.

With advances in the computer technologies and the advent of the World Wide Web, there has been an explosion in the amount and complexity of digital data, they being generated, stored, analyzed and accessed. Much of this information is multimedia in nature, including digital images video, audio, graphics and text data. In order to make use of this vast amount of data efficient and effective techniques to analyze and retrieve multimedia information based on its content need to be developed. Everyday both military and civilian equipment generates giga-bytes of images. We can not access to or make use of the information unless it is organized to allow efficient browsing, searching and retrieval.

Suppose a large image database. The content-based searching means to find an apriori-specified number of images, which are similar with another given, image. A major problem in dealing with large image databases is the efficiency of retrieval. One of the important issues in achieving such efficiency is the design of a suitable indexing scheme. Content-based image retrieval can be only effective if it entails reducing the search from a large and unmanageable number of images to a few that the user can quickly browse. The idea is to work with descriptions based on properties that are inherent in the images themselves. The idea behind this is that the natural way to retrieve visual data is by query based on the visual content of an image: the patterns, colours, textures, and shapes of image objects, and related layout and location information.

The motivation of this process is in close relations with the capacity of the solid human eye of accomplishing discrimination of images based on textures. The system performs multichannel decomposition using different orientations of the Gabor filters for a good discrimination.

The area of applications is very wide: multimedia documents, transaction systems, medical applications, query and browsing.

THE ARHITECTURE OF MULTIMEDIA APPLICATION MODEL

In the multimedia applications model architecture are four modules having a very strong interaction. Every module is composing by other small modules, and these small modules can be used very easy in other applications (Mocofan 2001).

I use four categories of modules with the application kernel. The application kernel, generally is a specific part for a multimedia application, it `s not possible to reuse the code, and for users is a close part (don't accept modifications). This kernel is "the manager" of entire application. It uses the functions, procedure, routines which are stored in modules. In many cases, the module is the equivalent of a functions library. One of the kernel jobs is to verify and to validate all the new functions implemented by the users. The architectures modules categories are:

- Objects processing functions;
- User interactions;
- Input/Output Operations;
- Integrated technologies.

Two main categories are included in *Objects processing functions*: temporal processing (the temporal aspect of objects is used in processing) and objects processing (for this category are two types of objects processing: single object processingand multiple objects processing).

In a multimedia application, we run in the same time

different sounds, video sequences and animations. The applications run correctly when are free hardware resources. It is necessary to manage very well the application and the hardware resources, this can be realized by many temporal references. The *Time_Line* routines must be very simple and easy to use (for a user with medium knowledge in the field of programming).

The main jobs of the *Time_Line* routines are:

- Management for time constraint objects and for the hardware resources used for playing that.
- Verification if the hardware resources are free to use them. If they are not free for use, it is necessary to stop playing the current object.
- Temporal references list construction for effects and transitions between objects.
- Management of temporal references jumping in collaboration with sub-modules *GUI* and *Events*.

Here, it is useful to use the aspect programming techniques. The very good results are mixing the object oriented programming techniques for multimedia objects description and aspect programming techniques for temporal management of the events.

Object Processing Functions – for reducing the space that a multimedia application requires in the context of increasing the power of calculation, I propose the utilization of "on-line processing" concept. That supposes the processing of the objects are made during the running of application. So, it can be used only one physical object in many ipostasies (in several scenes it can wear a variety of forms - it can be scaled, it can be rotate or deformed, etc.) during the running of an application. If the application uses a very large number of objects which processed by the same methods, handling these objects could be simplified using the idea of processing directly the information and not store it. We can meet two situations: the new generated object can be stored (the handler can personalize the multimedia objects starting from a model object) or the object can be destroyed after utilization (if we need it again the object can be regenerated).

If the processes are very complex, there is the risk of loosing a lot of time with calculation, and the result can not be obtained instantaneous. The processing functions have a big diversity and depend of the element that is processed. Sometimes its implementation can be very heavy. I recommend structuring of the elements, in a library that contains processing functions, which can be accessed later. For users, can be allowed the personalization of these functions, this is realizable by modification of the processing parameters.

Also is good that for experimented developers of multimedia applications to exist instructions on the way to implement new functions, which later can be distributed in new ways of multimedia applications area.

User Interaction - module is characterized through a graphical interfaces and events generated from interactions between application and user.

GUI – the graphical interface with the user is made from different graphical elements, objects which offers interactivity between user and application (buttons, links, images, little animations, etc.). These objects have attached routine, which start when one defined event appears in a library of functions, methods and events. The system events modify the way of application run, depends on the actions made by the user in the application. This makes an event, and for them, we will have a specified action. It is very important that we define correctly the system of events that might appear and the action whiches are realized after this.

It is necessary that we define one hierarchy of priorities in execution: in case of mixing over the events in the same time or in the case in which one event might provoke more than one action;

Input-Output Operations contains also one component which permits the work over network, the development of client/server services and a system of user validation. This module represents the way of interaction between the application and the out world (especially with the hardware resources of the computer system on which the application runs). We can found these functions in all the programming languages. I propose the writing of small routines to verify the peripheral state before their utilization, the release of some occupied resources because of some damage and the determination of peripheral performances.

The functions from *QoS* library are used to assure a good run of a multimedia application on a certain system, it is imperious the introduction of a routine for error detection that appears in the functionality. These functions will generate messages that will inform the user on the eventual problems that might appear and proposes the remove of problems.

My recommendation, for the *Network* routines, is to have possibilities to verify the network level of traffic. It is necessary to inform on the user about this level of traffic and to propose different solutions (if are some problems). Many applications use for communication implementation of client/server services. These kind of technologies are useful for job distribution in a heterogenic network, to enhance the speed of processing. If an application is accessed by many users in the same time or sequential, the management of users will do by specialized routines.

Integrated technologies module contains libraries dedicated to the work with databases (ODBC), integration libraries of the objects realized with other applications (OLE technologies), implementation of different kinds of services client-server libraries (DDE), optimizated functions for different equipments inside PC (MCI functions for sound, video and CD-ROM); evaluation libraries of quality service offered by the application on hardware resources by the user; function libraries which allowed searching object and resources which are used in the multimedia application; etc.

The biggest problems in multimedia applications are the synchronization constraints of different types of media, which runs simultaneous. I propose a few parameters which describe the quality of a service (of the way the program runs in a hardware environment) to be verified and respected by the programmer. These parameters are: the rate of application runs speed, the level of hardware utilization, the level of jitter, the temporal cumulate errors and the rate of environment errors;

The multimedia applications model architecture is presented in Figure 1.

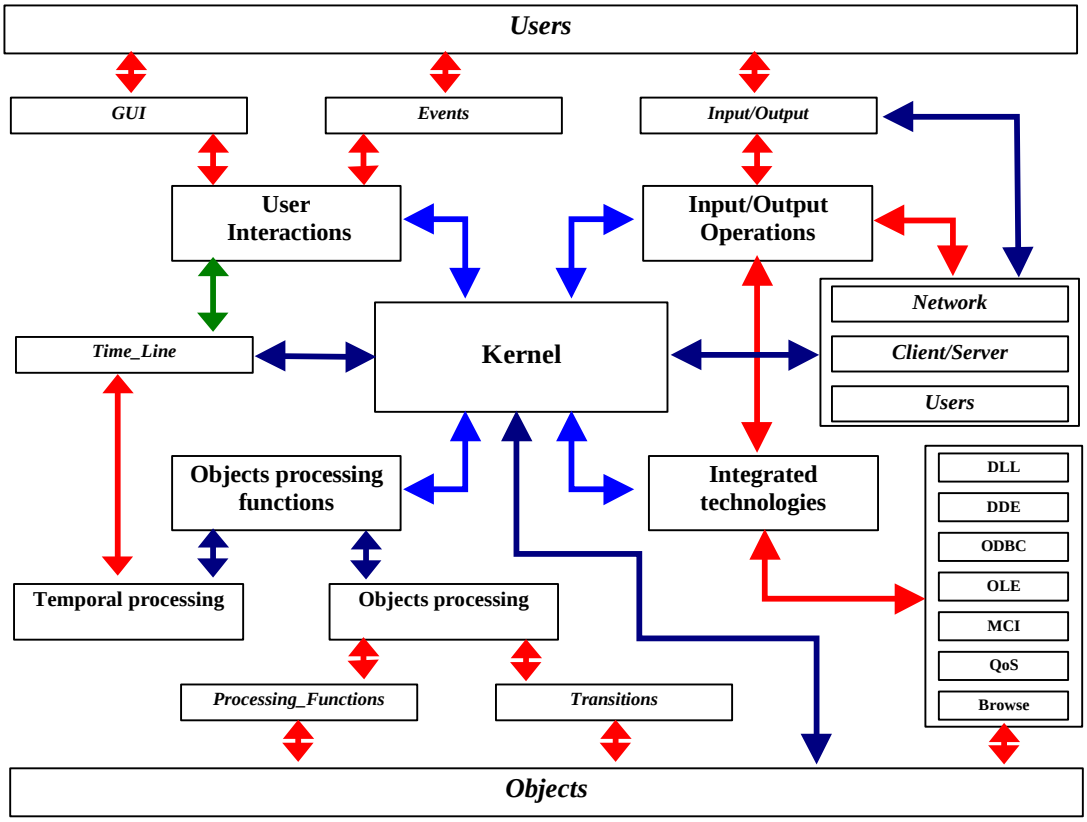

Figure 1. Multimedia application model architecture

MULTIMEDIA OBJECT DESCRIPTIVE MODEL

Each multimedia object has a value, a logical structure and an interpretation of the content for that value. The value of a multimedia object, his logical structure and the model of representation and storages must be described in the model of the multimedia object.

The multimedia describing model contains information that is referred to the logical composition of this object, the ways of synchronization and temporization for the components and the necessary parameters for displaying [lucrare Toma/ Mocofan].

The model of interpretation has to describe exactly the real world through the sub-objects that are composing it and through the relationship between them and each sub-object and the real world.

One multimedia object may be composed with other multimedia objects or with simple elements (un-composed multimedia objects). The multimedia object may be composing with the elements:

▪ Texts	▪ Images
▪ Sounds	▪ Videos
▪ Graphics	▪ Animations
▪ Scripts	

Table 1

The model of description can be synthesized like this:

- Each multimedia object is composed by a number of simple or complex sub-objects, which is organized sequentially, parallel or in a mix mode.
- The elements from the object structure can take a value, which can define an instance of the object. It is unnecessary for the object to have defined values for all of its components. We use the heritage concept from object oriented programming.
- For a more real description of the real world it can be established a semantic interpretation for each component of a multimedia object.
- The semantic description became usefully in the case of searching different objects in a multimedia application or when the number of objects is very large. It is necessary to have a structure of the elements based on semantic concepts.

Next, I will present the parameters that are considered very important to me for describing a multimedia object and simple elements. I have organized object descriptions in a structure of classes that has close properties to the classes that we meet in the case of object oriented programming.

For an efficient description of multimedia objects I proposed a describing parameters structure:

1. General parameters – that are describing the object, like object ID, storage physical location of the object, a flag which indicates if the object is simple or complex and other information about element structure of the object.
2. Semantic parameters – which describe the object through key words.
3. Type objects parameters – these parameters describe the object, its dimensions, the way of compression, the application that generated the object (useful for example in: the number of colors, dimensions, resolution, images, etc.).

4. Global parameters – which describe the content of objects throught many features. These parameters are useful in content database searching.
5. User parameters – specified by each user and useful only in some applications. After describing the object it follows the description of the primary elements which composed the object.

The general parameters of description of the primary elements are:

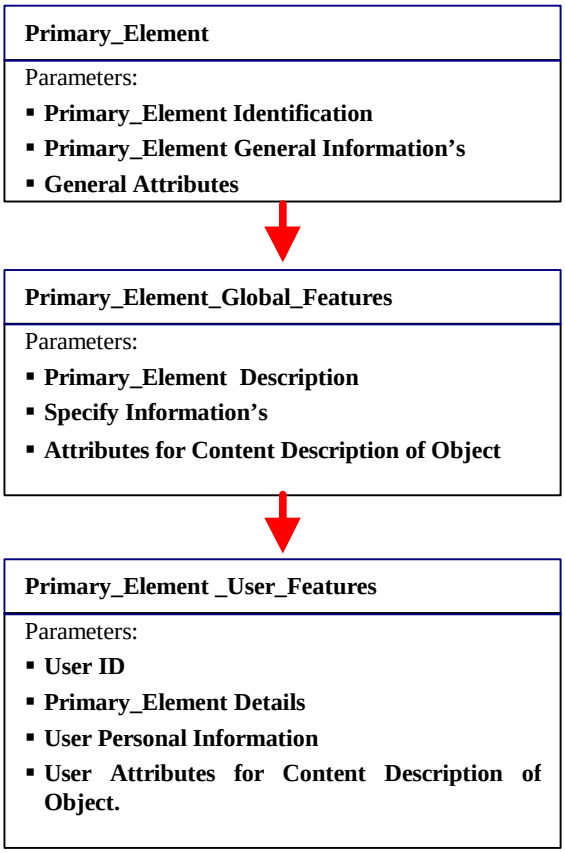

Figure 2. Primary Element Description

The general characteristics of this model consist in:

- the possibility to reuse the elements developed previously;
- the possibility for any user to introduce new functions;
- possibility to develop an old application;
- higher level of portability.

In concordance with the actual needs, my model present:

- Optimizations from the structure point of view, which is open for an ulterior development. I use them for the primary level of structure modules. For all modules it is possible to define new sub-modules. A sub-module accepts functions, methods and routines, it is possible to grow the number of that in the future (The user can add new functions, methods and routines).
- It is possible to reuse some parts of an old application.
- Object oriented implementation for structure of used objects in applications.
- Aspect oriented implementation for the temporal aspect of multimedia objects and applications.
- If the developers of multimedia environment use the proposal model, it is possible for the users to develop the functions libraries and to transfer these between applications.
- The model proposes a strong system for the quality of services. A special module QoS is developed for this.
- For the objects storage, can be use a standard database using a generalization of the ODBC concept.
- This model realizes a dynamic content for multimedia applications. The pages included in applications are generated dynamic from a database (in real time), these pages exist if the application runs, and are generated in concordance with the user options.
- Using the client/server technology, many users' acces the application in the same time using remote commands.

I propose to include in every multimedia environment tools for creation of new functions library. In this way it is possible to develop the multimedia software environment.

IMPLEMENTATION OF THE PROPOSED MULTIMEDIA APPLICATION MODEL FOR AN IMAGE DATABASES

One implementation for the proposed model is a database for multimedia objects. This database contains algorithms for content image database search.

Image Characteristics

Texture Gabor filtering - the method is based on a set of features, which operate in parallel during image decomposition into a collection of sub images. Individual filters are designed in such a way that these simultaneous actions should concentrate on a frequency area and on local spatial interaction. Because image texture is viewed differently according to analysis resolution, segmentation techniques that use Gabor filters based on a single frequency are not entirely satisfactory from the quality retrieval point of view (Trygve 1997). That is the way for an indexing method that uses a multichannel filtering technique being proposed, frequency space division being made according to perception by human visual system. It is very hard to go throughout the entire frequency space and for different orientations from the calculation point of view. That is why it is proposed for decomposition only through few channels. The system output is optimized from the point of view of average square error and takes into consideration 25 Gabor filters outputs (Mocofan 2000).

The co-occurrence matrix features - from practical observation results, generally, 5 indexes are enough to discriminate textures: homogeneity, contrast, entropy, correlation and local homogeneity. In my experiments on different textures allowed me to obtain textures' classifications using just a few of characteristics, i.e., a low calculation complexity.

The global image features - global colour distribution is one of the most important and useful features, when in image we have many textures and details. In this case is difficult to use segmentation algorithms and the global features are significant. The colours distribution representations are the colour histograms and represent with success global features. The histograms represent generally the image, and the composition, orientations or translations of objects are not important for them. The histograms can be compared by an intersection operation (Cocquerez et al 1995).

If we use colour histograms for the indexing methods, is not possible to search in small area from image because the histograms are only for the entire image and is not possible to split histograms data for user areas.

The drawbacks of these methods can be reduced by the splitting of the original image in 4 or 9 parts. For every part we calculate the colour histograms. The volume of features data grows very fast, each new small part will introduce the same data like the original image.

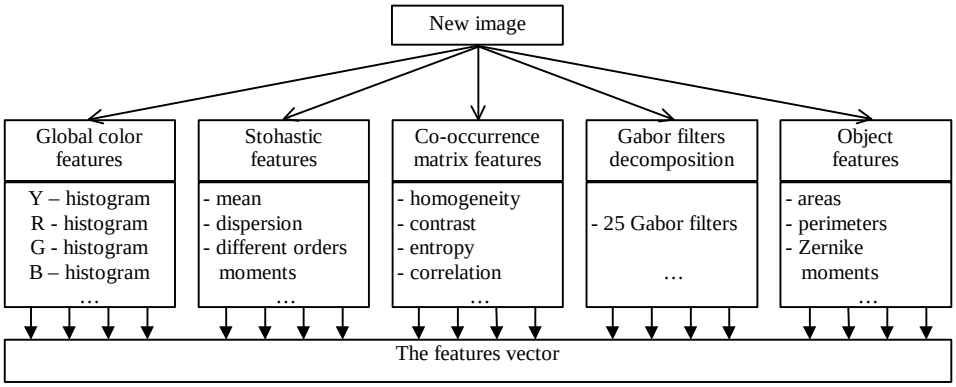

Figure 3. The categories of features used in the indexing process

The Used Tree Searching Algorithm

The goal of this paper is to find a similarity between images, and to clustering the most closed images in concordance with the human visual system. For that reason are used the global features (colour histograms), the stohastic features (mean, dispersion, different moments), the co-occurrence matrix features (homogeneity, correlation, entropy, contrast) and the image decompositions of 25 Gabor filters (these filters are a good model for the human visual system).

I tested the follow methods for discrimination (Mocofan 2002):

- the Euclidian distance features;
- the Euclidian distance between histograms;
- the similarity between histograms;
- a method with MLP neural networks;
- a method with thresholds for all the features;

There is a big problem; the time spending for distance evaluation is to long. I propose a tree algorithm which uses five features (the most important features) from the five categories presented in this paper.

First step is to extract all features for the query image. We obtain a features vector with the follow structure $F_{1,1}^{db} \dots F_{1,j}^{db}$, $F_{2,1}^{db} \dots F_{2,l}^{db}$, $F_{3,1}^{db} \dots F_{3,u}^{db}$, $F_{4,1}^{db} \dots F_{4,v}^{db}$, $F_{5,1}^{db} \dots F_{5,m}^{db}$. $F_{l,j}^{qi}$. Where, *l* represent the categories of features, *j* represent the order of features in this category. *qi* represent the query image and *db* image from databases.

The second step is to verify if the distance between the most important features are under a threshold q_l.

Mixing the trees techniques with the human interpretation of features the speed of the searching process increases. There are some important features for the human visual system.

The first feature tested is the mean. Is a very good global feature but not sufficient. If the distance between the feature of query image and the feature of an image from databases is under a threshold q_l the algorithm will go to the next feature from other categories. If this condition is not satisfied, the algorithm will jump to a new image from databases.

The second feature tested is the histograms for the luminance signal. The histograms are global features for an image and it is possible to obtain a very good similarity for two different images but with the same histograms. For example an image with vertical black and white bars and an image with horizontal black and white bars are two distinct images, but, the histograms can be the same.

From the co-occurrence matrix features, the first feature used is the entropy.

For objects searching I propose the rate area/perimeter and from Gabor filters can be used the horizontal and vertical orientations.

The distance between the query image features and the image from databases can compared using different imposed thresholds. If one of them is not respected the algorithm will jump to evaluate other image from databases. After the evaluations of most important features, all features will be used for a fine definition of the searching process.

This algorithm increases the speed of searching. For large image databases the time necessary to find the image will be very short. For example, the mean, and few orientations of the Gabor filters are representative (the horizontal and vertical direction, the diagonals). If the restrictive condition for this feature is not satisfied, the search process is stopped and other images features are compared. The computational volume will be reduced.

Experimental Results

In my simulations I use for discrimination the Euclidian metrics between features and between histograms, the similarity between histograms, a MLP neural network and the proposed threshold method.

The algorithm was tested with 300 images from a database. The spectrum of this database was very large: real images from nature, animals, peoples, textures, artificial images.

After many experiments some conclusions could be extracted:

- The image decomposition by Gabor filters is a similar approach with the visual human interpretation system. It's not necessary to use many filters. Using a large number of filters the computational volume will grow, and the algorithm will be very slowly.
- An important step in the algorithm design is to make a very right decomposition with Gabor filters, to cover the entire space.
- A large range for threshold is not representative for clustering images. Using an iterative system, the threshold can be reduced step by step. The resulted images are more and more similar. A very small threshold is not a solution, because the number of features used is also too small and only few features don't describe very well an image.
- If we choose only few Gabor filters, with directivity properties, the search process results are images with a small similarity, only few properties are the same.
- The Gabor filter decomposition represent a very good features set, but the co-occurrence matrix offer usefully features like: contrast, homogeneity, entropy and correlation.

Figure 4. The interface of MMDB:NICE

The similarity for the human visual interpretation system is collections of features which have a little tolerance. The global interpretation of image, like similarity between histograms, is useful, but it is not a sufficient method.

An index in a database consists in a collection of entries, one for each data item, containing the value of a key attribute for that item and a reference pointer that allows immediate access to the item. Selection, derivation, and computation of image features that provide more useful query expressiveness are the main goal for this work. I don't implement the methods that are based on exact matching; I use an implementation of retrieval methods based on similarity to obtain clustered images.

CONCLUSIONS

In these days, it is very important to have an open environment for multimedia applications development, because are many application developers with a good experience in this field, and this experience can be used for the functions library development.

From design point of view, the time necessary to realize a good concept and the modeling for application is much longer. From implementation point of view, the time for creation is in this case, much shorter. The global time spend for realization of an entire multimedia application is generally shorter. If it uses the real time content generation concept for pages, the time of physical implementations degreases very well.

For example, to realize a multimedia presentation for a company which produce hundred goods it is necessary to make hundred pages, one page for every product. If all the information about the product is stored in a database, the implementation task for all pages is to make the model for a single page, and after that to connect this page to database, it is not necessary to realize hundred different pages. In this dynamic way the reduction of the implementation time is considerable. More, the size of the executing file is smaller, because the object are loaded only when are used.

This paper presents a new interactive multimedia application model which combines the advantages of objects oriented programming with aspect programming. The model proposes a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases. Using this model it is possible to develop very performant multimedia applications.

REFERENCES

Cocquerez J. P., Philipp S. – "Analyse d'images: filtrage et segmentation", Mason, Paris, 1995

Mocofan M., Pescaru D., "A parallel implementation of a algorithm for content-based image indexing and retrieval using texture features", *Proceedings of International Conference "Communications 2000"*, 7 - 9 December 2000, Bucharest, Romania, pp. 85-88, 2000.

Mocofan M., Pescaru D. – "Aplicatii interactive multimedia. Programarea OpenScript", Ed. Politehnica, Timisoara, 2001.

Mocofan M., Vasiu R. – "Algorithm based on image similarity for content search in an image database", *THIRD COST #276 WORKSHOP on Information and Knowledge Management for Integrated Media Communication*, 11- 12 October, 2002, Budapest, Hungary.

Trygve R. – "Filter and Filter Bank Design for Image Texture Recognition", *PHD Thesis* 1997.