

14. ТЕЛЕКОМУНИКАЦИОНИ ФОРУМ  
*14th TELECOMMUNICATIONS FORUM*

# ТЕЛФОР 2006

## *TELFOR 2006*



**ПРОГРАМ**  
***PROGRAM***

Београд, 21., 22. и 23. новембар 2006. године  
*Belgrade, 21, 22 and 23 November 2006*

Сава Центар  
*THE SAVA CENTER*

## **STEERING COMMITTEE - TELFOR 2006**

**Prof. dr George Paunović, TS, ETF, Beograd, Chairman**  
Mr Ivan Đurović, Telekom Srbija  
Mr Mihailo Jovanović, JP PTTS Srbija  
Prof. dr Vladimir Katić, Chairman, IEEE S&M Section, FTN, Novi Sad  
Prof. dr Branko Kovačević, dekan ETF, Beograd  
Mr Siniša Davitkov, DT, IRITEL, Beograd  
Prof. dr Zoran Petrović, ETF, Beograd  
Docent dr Nataša Nešković, Treasurer, IEEE S&M Section, ETF, Beograd

## **PROGRAM COMMITTEE - TELFOR 2006**

**Prof. dr Zoran Petrović, ETF, Beograd, Chairman**  
Prof. dr Vladanka Aćimović-Raspopović, SF, Beograd  
Prof. dr Nataša Gospić, SF, Beograd  
Prof. dr Vlado Delić, FTN, Novi Sad  
Prof. dr Dušan Drajić, ETF, Beograd  
Prof. dr Miroslav Dukić, ETF, Beograd  
Prof. dr Antonije Đorđević, ETF, Beograd  
Prof. dr Miroslav Lutovac, ETF, Beograd  
Prof. dr Aleksandar Marinčić, ETF, Beograd  
Prof. dr Ljiljana Milić, IMP, ETF, Beograd  
Docent dr Aleksandar Nešković, ETF, Beograd  
Docent dr Nataša Nešković, ETF, Beograd  
Prof. dr Zorica Nikolić, EF, Niš  
Prof. dr George Paunović, ETF, Beograd  
Prof. dr Predrag Pejović, ETF, Beograd  
Prof. dr Miodrag Popović, ETF, Beograd  
Prof. dr Jelica Protić, ETF, Beograd  
Prof. dr Branimir Reljin, ETF, Beograd  
Docent dr Irini Reljin, Viša ICT škola, ETF, Beograd

## **SCIENTIFIC COMMITTEE – TELFOR 2006**

Akademik prof. dr Ilija Stojanović  
Prof. dr Ljiljana Milić, IMP, ETF, Beograd  
Prof. dr Dušan Drajić, ETF, Beograd  
Prof. dr Vladanka Aćimović-Raspopović, SF, Bgd.  
Prof. dr Branka Vučetić, Univ. Sidnej, Australija  
Prof. dr Nataša Gospić, SF, Beograd  
Mr Siniša Davitkov, IRITEL, Beograd  
Prof. dr Vlado Delić, FTN, Novi Sad  
Prof. dr Miroslav Dukić, ETF, Beograd  
Ljiljana Đorđević, Telekom Srbija  
Milivoje Žunić, IRITEL, Beograd  
Doc. dr Milan Janković, RATEL  
Dragan M. Kovačević, JP PTTS Srbija  
Dragan Đ. Kovačević, Telekom Srbija  
Prof. dr Đemal Kolonić, ETF, Banja Luka  
Prof. dr Ivo Kostić, ETF, Podgorica  
Prof. dr Slobodan Lazović, SF, Beograd

Vladimir Lučić, Telekom Srbija  
Akademik prof. dr Aleksandar Marinčić  
Slobodan Milosavljević, Telekom Srbija  
Prof. dr Bratislav Milovanović, EF, Niš  
Prof. dr Vladimir Milošević, FTN, NS  
Zorica Nestorović, Telekom Srbija  
Doc. dr Aleksandar Nešković, ETF, Beograd  
Doc. dr Nataša Nešković, ETF, Beograd  
Prof. dr Zorica Nikolić, EF, Niš  
Prof. dr George Paunović, ETF, Beograd  
Prof. dr Milica Pejanović-Đuriš., ETF, CG  
Prof. dr Grozdan Petrović, RATEL, ETF, Bgd  
Prof. dr Zoran Petrović, ETF, Beograd  
Dr Predrag Petrović, IRITEL, Beograd  
Prof. dr Jovan Radunović, RATEL, ETF, Bgd  
Prof. dr Branimir Reljin, ETF, Beograd  
Mr Momčilo Simić, MKI, RTS, Beograd

## **ORGANIZING AND TECHNICAL COMMITTEE - TELFOR 2006**

**Prof. dr Ljiljana Milić, ETF, IMP, Beograd**  
Prof. dr Dušan Drajić, ETF, Beograd  
Prof. Dr Miroslav Lutovac, IRITEL, ETF, Beograd  
Danka Despotović, ETF Beograd  
Ivan Nador, Telekom Srbija  
Danka Milenković, JP PTTS Srbija  
Dragoljub Poposki, Muzej PTT  
Branko Šojić, DT, Beograd  
Mr Nenad Krajnović, ETF, Beograd  
Mr Jelena Čertić, ETF, Beograd  
Mr Milan Bjelica, ETF, Beograd  
Marija Nikolić, INI Ltd., Beograd

### **PRESS & MEDIA**

Tanja Blečić, Telekom Srbija  
Zorica Panić, JP PTTS Srbija  
Danka Despotović, ETF, Beograd

### **RECENZENTI**

Aleksandra Smiljanić  
Bratislav Milovanović  
Branka Jokanović  
Dejan Popović  
Dragiša Krstić  
Dragan Milićev  
Dragan Vasiljević  
Dragan Kovačević  
Dušan Drajić  
Dragan Bogojević  
Antonije Đorđević  
Miljko Erić  
Miodrag Gmitrović  
Stevica Graovac  
Dejan Gvozdić  
Irina Reljin  
Zoran Jakšić  
Jovan Đorđević  
Slobodan Jovičić  
Branko Kolundžija  
Nenad Krajnović  
Milan Janković  
Ljubiša Stanković  
Miroslav Lutovac  
Petar Matavulj  
Milorad Tošić  
Milorad Obradović  
Čedomir Milosavljević  
Mladen Mileusnić  
Milo Tomašević  
Nataša Gospić

Nataša Nešković  
Boško Nikolić  
Aleksandar Nešković  
Milica Naumović  
Đorđe Paunović  
Predrag Pejović  
Miodrag Popović  
Božidar Radenković  
Ninko Radivojević  
Nikola Rajaković  
Branimir Reljin  
Aleksandar Rodić  
Dragutin Salomon  
Emil Šećerov  
Dušan Starčević  
Mirjana Stojanović  
Dejan Tošić  
Lazar Trpković  
Vladanka Aćimović-Raspopović  
Vujo Drndarević  
Vlado Delić  
Vesna Ivković  
Vladimir Petrović  
Desimir Vučić  
Željen Trpovski  
Borislav Odadžić  
Zorica Nikolić  
Zoran Jovanović  
Zoran Petrović  
Zoran Bojković  
Zlatan Stojković

# Searching of images stored in a database using content and pixel based methods

Igor Stojanovic, Momcilo Bogdanov, *Member, IEEE*, and Sofija Bogdanova, *Member, IEEE*

**Abstract** — In the paper we consider combined pixel and content based searching of images. We propose an application of Stone's method of progressive wavelet correlation using Fourier methods for pixel-based searching of images stored in a database. The proposed interface between the Matlab workspace and the database is described. The Oracle Database and the IBM QBIC are used for investigation purposes.

**Keywords** — Database, JPEG, wavelets.

## I. INTRODUCTION

Image retrieval and multimedia systems are changing our life-styles because they together cover creation, maintenance, accessing and retrieval of video, audio, image, textual and graphic data. Research on multimedia systems and image retrieval has gained tremendous momentum during the last decade. This is due to the fact that multimedia databases cover the text, audio, video and image data which help us receive enormous amounts of information and which have brought fundamental changes in our life style. Images are being generated at an ever-increasing rate by various sources for military purposes, aviation satellites, biomedical purposes, scientific experiments and home entertainment. Application areas in which image retrieval is a principal activity are numerous and diverse: art galleries, engineering design, scientific database management, weather forecasting, to mention only a few of them.

The fastest methods available today use descriptor-based search techniques. IBM QBIC ([www.qbic.almaden.ibm.com](http://www.qbic.almaden.ibm.com)) [1] is an example of this type of search engine. Images with higher information content, such as satellite images and medical images, are difficult to encapsulate with descriptors. Normalized correlation coefficients, an instance of pixel-based search techniques, measure the differences between images and patterns [2].

The paper is organized as follows. The method of Stone [2] for computing progressive wavelet correlation is outlined in Section II. Our proposal concerning an application of this method for searching of images stored in a database is presented in section III. Section IV covers

some of the work done in the field of content-based image retrieval and combination of content and pixel based approaches.

## II. STONE'S METHOD OF PROGRESSIVE WAVELET CORRELATION

### A. Overview of the method

The basic ideas of Stone's method include elimination in the earlier phases of searching, correlation in the frequency domain for fast computation, DCT in factorizing form, and wavelet representation of signals for efficient compression [2].

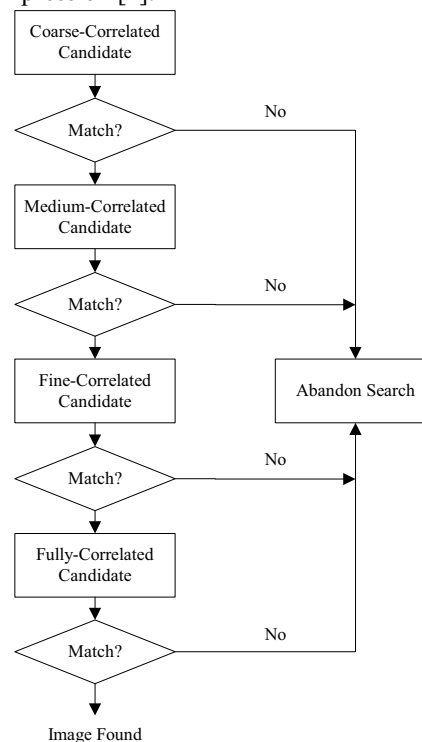


Fig. 1. Flow diagram of Stone's method

The algorithm consists of three incremental steps:

1. *Coarse correlation* – every eighth point of the correlation is generated.
2. *Medium correlation* – obtain the correlation at indices that are multiples of  $4 \bmod 8$  of the full correlation.
3. *Fine correlation* – obtain the correlation at indices that are multiples of  $2 \bmod 8$  and  $6 \bmod 8$  of the full correlation.
4. *Full correlation* – obtain the correlation at odd indices.

Fig. 1 is a flow diagram showing the steps performed for an image search according to the method of Stone.

I. Stojanovic, ICT Sector, Customs Administration of RM, Lazar Licenski 13, 1000 Skopje, Macedonia; (phone: +389 2 3293938; e-mail: [igor.stojanovic@customs.gov.mk](mailto:igor.stojanovic@customs.gov.mk)).

M. Bogdanov and S. Bogdanova, Faculty of Electrical Engineering, Karpos II b.b., P.O. Box 574, 1000 Skopje, Macedonia., (e-mail: [bogdanov@etf.ukim.edu.mk](mailto:bogdanov@etf.ukim.edu.mk), [sofija@etf.ukim.edu.mk](mailto:sofija@etf.ukim.edu.mk)).

### B. Two-Dimensional Case

We assume that the image size is  $N$  by  $N$ . In step 1, we have 64 subbands of length  $N^2/64$ . We perform one step of the inverse 2D JPEG transfer function, and one 2D step of the forward Fourier transform function. The next step includes adding the 64 subbands point by point to create a 2D array of size  $N/8$  by  $N/8$ . Taking the inverse Fourier transform, we obtain the correlations at points that lie on a grid that is coarser than the original pixel grid by a factor of 8 in each dimension. In step 2, we obtain 16 subbands of size  $N^2/16$  by adding the 16 subbands point by point, and taking the Fourier inverse. We will obtain the correlation values on a grid that is coarser than the original grid by a factor of 4 in each dimension. In step 3, we obtain 4 subbands of size  $N^2/4$ . Finally, in step 4, the full resolution is obtained.

Formulas for calculating normalized correlation coefficients that measure differences between images and patterns are given in [2]. Normalized correlation coefficients can be computed from the correlations described above. The normalization is very important because it allows for a threshold to be set. Such a threshold is independent of the encoding of the images.

The normalized correlation coefficient has a maximum absolute value of 1. Correlations that have absolute values above 0.9 are excellent, and almost always indicate a match found. Correlations of 0.7 are good matches. Correlations of 0.5 are usually fair or poor. Correlations of 0.3 or less are very poor. There is a tradeoff between the value of the threshold and the likelihood of finding a relevant match. Higher thresholds reduce the probability of finding something that is of interest, but they also reduce the probability of falsely matching something that is not of interest.

## III. ADAPTATION OF STONE'S METHOD FOR SEARCHING IN A DATABASE

### A. Image Store and Matlab Database Toolbox

The method of Stone provides guidelines on how to locate an image in the image library. To make this method practical, we must first decide how to store the images. The initial choice is to store them in a disk file system. This can be seen as the quickest and simplest approach. A better alternative that should be considered is to store those images in a database. In the past five years, with changes in database technology and improvements in disk performance and storage, the rules have changed and it now makes business sense to use the database to store and manage all of an organization's digital assets. Databases offer several strengths over traditional file system storage, including manageability, security, backup/recovery, extensibility, and flexibility.

We use the Oracle Database for investigation purposes. There are two ways of storing an image into the Oracle Database. The first one is the use of Large Objects – LOB, and the second one is the use of Oracle *interMedia*.

Unstructured data such as text, graphic images, still video clips, full motion video, and sound waveforms tend

to be large in size. A typical employee record may be a few hundred bytes, while even small amounts of multimedia data can be thousands of times larger. Datatypes that are ideal for large amounts of unstructured binary data include the BLOB datatype (Binary Large Object) and the BFILE datatype (Binary File object).

Oracle *interMedia* is a feature that enables Oracle Database to store, manage, and retrieve images, audio, video, or other heterogeneous media data. *interMedia* uses object types, similar to Java or C++ classes, to describe multimedia data. These object types are called ORDAudio, ORDDoc, ORDImage, and ORDVideo. An instance of these object types consists of attributes, including metadata and the media data, and methods.

To store images into the database we use the BLOB datatype. After creation of one BLOB column defined table we also create a PL/SQL package with loading of images procedure (load named) included. This procedure is used to store images into the database.

The implementation of the method of Stone into Matlab and connection of the algorithm with the database are the next steps. The Database Toolbox is one of an extensive collection of toolboxes for use with Matlab. The Database Toolbox enables one to move data (both importing and exporting) between Matlab and popular relational databases. With the Database Toolbox, one can bring data from an existing database into Matlab, use any of the Matlab computational and analytic tools, and store the results back in the database or in another database. The Database Toolbox connects Matlab to a database using Matlab functions. Data can be retrieved from the database and stored in the Matlab workspace. At that point, the extensive set of Matlab tools can be used to work with the data. Database Toolbox functions can be included in Matlab M-files. To export the data from Matlab to a database, Database Toolbox functions can be used. The Visual Query Builder (VQB), which comes with the Database Toolbox, is an easy-to-use graphical user interface (GUI) for exchanging data with the database. The VQB can be used instead of or in addition to Database Toolbox functions.

Before the Database Toolbox is connected to a database, a *data source* must be set. A data source consists of data for the toolbox to access, and information about how to find the data, such as driver, directory, server, or network names. Instructions for setting up a data source depend on the type of database driver, ODBC or JDBC.

For Windows platforms, the Database Toolbox supports Open Database Connectivity (ODBC) drivers as well as Java Database Connectivity (JDBC) drivers. For UNIX platforms, the Database Toolbox supports Java Database Connectivity (JDBC) drivers. An ODBC driver is a standard Windows interface that enables communication between database management systems and SQL-based applications. A JDBC driver is a standard interface that enables communication between Java-based applications and database management systems. The Database Toolbox is a Java-based application. To connect the Database Toolbox to a database's ODBC driver, the toolbox uses a

JDBC/ODBC bridge, which is supplied and automatically installed as part of the MATLAB JVM. Figure 2 illustrates the use of drivers with the Database Toolbox.

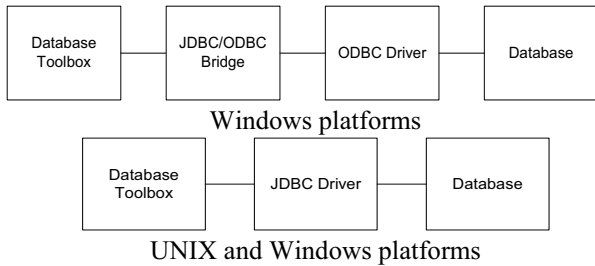


Fig.2. The use of drivers with the Database Toolbox

If the Windows-based database supports both ODBC and JDBC drivers, the JDBC drivers might provide better performance when accessing the database because the ODBC/JDBC bridge is not used.

The connection definition can be established using either the Oracle ODBC driver or the Microsoft ODBC driver for Oracle. To use these drivers, it is necessary to have Matlab and the Oracle client installed on the same computer. During testing we realized that Microsoft ODBC drivers for Oracle cannot be used for tables with columns of data type LOB. For testing purposes JDBC drivers were usually used [3].

After setting up the data source for connecting to and importing data from a database we have used several standard functions of the Matlab Database Toolbox.

We can retrieve BINARY or OTHER Java SQL data types. However, the data might require additional processing once retrieved. For example, data can be retrieved from a MAT-file or from an image file. Matlab cannot process these data types directly. One needs knowledge of the content and might need to massage the data in order to work with it in Matlab, such as stripping off leading entries added by the driver during data retrieval.

For the purpose of saving the extracted data into file "testfile", we created the Matlab file parsebin.m. Using the "imread" function, we stored the file date into a two-dimensional output variable x.

In working with the Microsoft Access database, ODBC drivers are used for extracting data from the database. The extracting process must be adjusted by cutting the header created by the driver. Let "m" be quantity of bytes attached to the beginning of the data package by the ODBC driver. The quantity of bytes depends on the file type (file extension).

To discover the value of "m" we created a procedure in Matlab. That procedure helps us find the adequate value for "m" when linking with the image format.

When working with Oracle databases and extracting data with JDBC and ODBC drivers there is no need for adjustment, so "m" has always the value "1" ( $m="1"$ ).

### B. HTTP Application

The last step in the adaptation is to create Matlab applications that use the capabilities of the World Wide Web to send data to Matlab for computation and to display the results in a Web browser. The Matlab Web Server

depends on TCP/IP networking for transmission of data between the client system and Matlab.

In the simplest configuration, a Web browser runs on your client workstation, while Matlab, the Matlab Web Server (matlabserver), and the Web server daemon (httpd) run on another machine as shown in Figure 3.

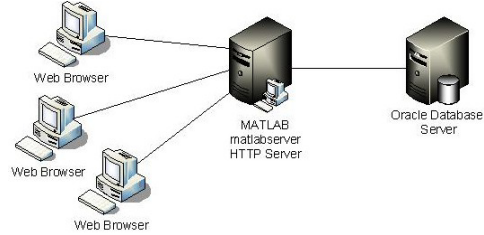


Fig. 3. The simplest configuration

In a more complex network, the Web server daemon can run on a separate machine (Fig. 4).

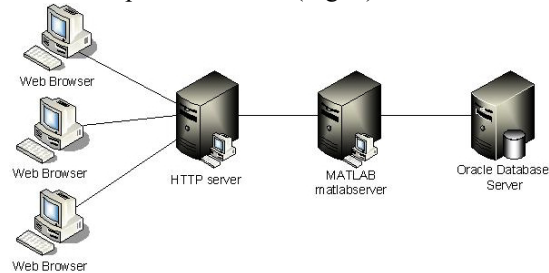


Fig. 4. HTTP server running on separate machine

The input mask of our application consists of three parameters: the image size  $N$ , the threshold  $lim$ , and the name of the image that we are looking for (Fig. 5).

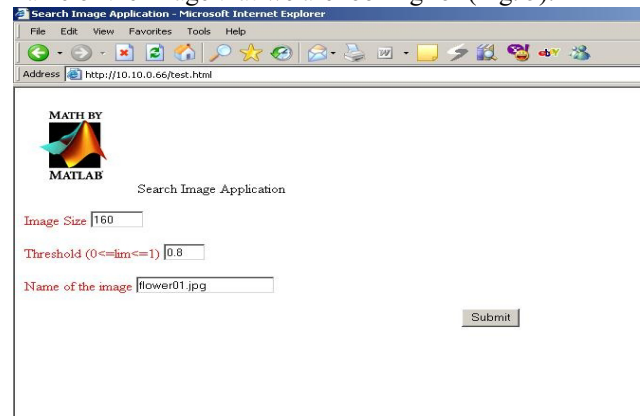


Fig. 5. Input Mask of our application

## IV. CONTENT-BASED IMAGE RETRIEVAL

### A. Background

Many research works have been published in the field of Content-Based Image Retrieval (CBIR). However, no universally accepted model has yet been developed. The research concentrated on image segmentation based on low-level features like color, shape, texture and spatial relations. To find the semantic meanings or high-level meanings of an image, like whether it is the image of human beings or a bus or a train and so on, is still a problem. Attempts are being made to link low-level and high-level features. However, it is proving difficult for the very simple reason that there remains a vast gap between

human perception and computer perception. Some of the CBIR models are QBIC (Query by Image Content), Virage, Pichunter, VisualSEEK, Chabot, Excalibur, Photobook, Jacob, and Digital Library Project.

The QBIC technology allows content-based queries. More precisely, it performs queries based on visual properties of the input image, such as color or texture.

Querying by content requires two phases:

- *Database creation*: A preprocessing step to compute numeric features for an image or set of images, and store them in a database.
- *Database query*: The run-time step that uses the computed features to find images similar to a query specification.

The database creation step uses one or more feature classes to compute the features of input images as numeric values. Each feature class creates a feature table in the database where these computed values are stored. During the database query phase, QBIC compares feature data in the query with the computed data in the feature tables. A query can search on one or more features for similarity.

A simple query involves only one feature. An example of a simple query would be to find images in the database that have a color distribution similar to the query image. A complex query involves more than one feature and can be either a multi-feature or a multi-pass query. In a multi-feature query, QBIC searches through different types of feature data in the database in order to find images that closely resemble the query image. All feature classes are treated equally during the database search, and all involved feature tables are searched at the same time. An example of a multi-feature query would be finding images in the database that have a color distribution and texture similar to a query image. In a multi-pass query, the output of an initial search is used as the input for the next search. QBIC reorders the search results from a previous pass based on the feature distances in the current pass. An example of a multi-pass query would be finding images in the database that have a color distribution similar to the query image, and then reordering the results based on color composition. For multi-feature and multi-pass queries, features can be weighted to specify their relative importance, which provides flexibility for advanced applications where the returned results must be fine-tuned.

QBIC queries may be of the following types:

1. *Query by example*: To locate in a database images that are similar to the “example” query image (whose feature data has already been computed by QBIC).
2. *Query by image*: May be invoked on images whose feature data has not been precomputed by QBIC. QBIC computes the feature data for the input image, and then compares it with data in database.

#### B. Combination of two methods

Content-based image retrieval is a bottleneck of the access of multimedia databases simply because there remain vast differences in the perception capacity between a human being and a computer [4].

The normalized correlation coefficients are impractical

to use during image processing owing to the big number of operations per image they entail. Current processor performance does not allow a practical implementation of this method. One alternative is to combine IBM QBIC [1] and the normalized correlation coefficients. This yields an improvement in the locating time for a given image in the library.

Using QBIC, we established a database for the following characteristics of images: color, text, color histogram, and texture feature. We query the database for those images in the library with the most similar characteristics to the input image. On this set of candidates, we apply the normalized correlation coefficients to obtain the desired image. As an example, we show our work on locating the image flower02.jpg in a database. Using QBIC, we isolated ten candidate images based on the Color Histogram Feature. After that these images are subjected to detailed pixel-based searching based on the normalized correlation coefficients.

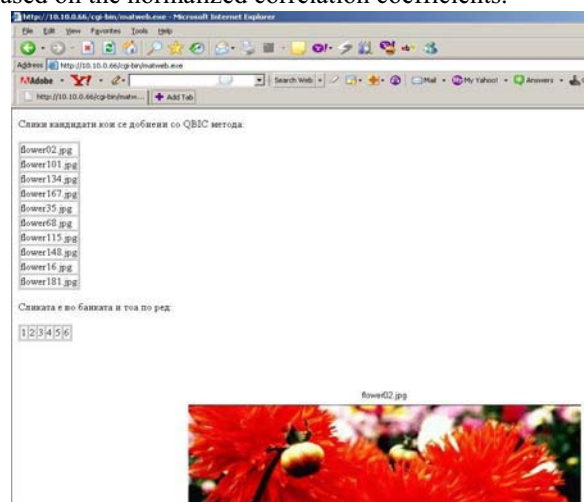


Fig. 6. Output mask of our application

It took 240 seconds to locate an image in a database of 200 images with normalized correlation coefficients. By combining the two methods, we reduced the waiting time to 40 seconds.

#### REFERENCES

- [1] M. Flickner et al., “Query by image and video content: The QBIC system,” *IEEE Comp.*, vol. 28, pp. 23-32, Sept.1995.
- [2] H. S. Stone, “Progressive Wavelet Correlation Using Fourier Methods,” *IEEE Trans. Signal Processing*, vol. 47, pp. 97-107, Jan. 1999.
- [3] I. Stojanovic, M. Bogdanov, “Pixel-Based Searching of Images Stored in a Database”, *ICEST2006*, pp. 165-168, Sofia, Bulgaria, 29 June – 1 July 2006.
- [4] *Multimedia Systems and Content-Based Image Retrieval*, Sagarmay Deb, University of Southern Queensland, Australia, 2004.

#### SEARCHING OF IMAGES STORED IN A DATABASE USING CONTENT AND PIXEL BASED METHODS

Igor Stojanovic  
Momcilo Bogdanov  
Sofija Bogdanova