Content-Based Image Retrieval on Mobile Devices

Iftikhar Ahmad[1], Shafaq Abdullah[2], Serkan Kiranyaz[2], Moncef Gabbouj[2]
[1]Nokia Corporation. P.O.Box 88, FIN 33721, (Tieteenkatu 1, Tampere 33720, Finland)
Email: Iftikhar.ahmad@nokia.com, Phone: +358 503480631
[2]Signal Processing Laboratory, Tampere University of Technology, P.O.Box 553, FIN-33101
Tampere 33720, Finland. Email**:** shafaq.abdullah@tut.fi**,** serkan@cs.tut.fi, moncef.gabbouj@tut.fi

ABSTRACT

Content-based image retrieval area possesses a tremendous potential for exploration and utilization equally for researchers and people in industry due to its promising results. Expeditious retrieval of desired images requires indexing of the content in large-scale databases along with extraction of low-level features based on the content of these images. With the recent advances in wireless communication technology and availability of multimedia capable phones it has become vital to enable query operation in image databases and retrieve results based on the image content. In this paper we present a content-based image retrieval system for mobile platforms, providing the capability of content-based query to any mobile device that supports Java platform. The system consists of light-weight client application running on a Java enabled device and a server containing a servlet running inside a Java enabled web server. The server responds to image query using efficient native code from selected image database. The client application, running on a mobile phone, is able to initiate a query request, which is handled by a servlet in the server for finding closest match to the queried image. The retrieved results are transmitted over mobile network and images are displayed on the mobile phone. We conclude that such system serves as a basis of content-based information retrieval on wireless devices and needs to cope up with factors such as constraints on hand-held devices and reduced network bandwidth available in mobile environments.

Keywords: Content, retrieval, indexing, image, mobile.

1. INTRODUCTION

The mobile phone industry is going through phenomenal change over the past few years with significant advances in areas of communications and multimedia. 3G [20] services are already in the market and offer a great bandwidth to meet the rising demands of the users with delivery of high quality multimedia services. Currently state-of-the-art multimedia compliant mobile phones equipped with digital cameras and camcorders have inherent support for network connection and thus, enable access to large amount of digital media. The processing power and memory capacity of mobile phones are increasing all the time. Nowadays, mobile platforms offer rich programming APIs (Application Programming Interface) to developers. With the generation of digital media by capturing and storing facility in smart phones, there is a need for content management and system to provide rapid retrieval of digital media items from large archives. Therefore, it has become vital to retrieve desired information expeditiously and efficiently using these devices.

On the other hand, the content-based image retrieval area possesses a tremendous potential for exploration and utilization equally for researchers and people in industry due to its promising results. It has been an active area of research over a past decade. Rapid retrieval of desired multimedia content requires indexing of the content in large-scale databases containing media items along with extraction of low-level features based on content. The low-level features based on shape, color and texture are extracted from images using automated feature extraction algorithm. Systems such as "Multimedia Video Indexing and Retrieval System" (MUVIS) [16], [3], VisualSEEk [9], Photobook [7], Virage [21] have common feature of having a framework for indexing and retrieving images and audio-video files. The contemporary MUVIS has been developed as a system for content-based multimedia retrieval on a PC-based environment. MUVIS provides a unified and global framework and consists of robust set of applications for capturing, recording, indexing and retrieval combined with browsing and various other visual and semantic capabilities.

With encouraging results of content-based information retrieval, favorable mobile platform support and limitations imposed by text-based queries [4] researchers and scientists have undertaken challenge of meeting content management of users using mobile devices. Annotation of digital media at the time of capture or store is cumbersome for a mobile user, which has been the basis of metadata creation process [8]. Systems such as IDeixis [10] find application in content-based image retrieval for location based services but the users usually experience a large latency because of an inadequate mechanism for image transport which is based on Multimedia Message Service (MMS). In this paper, we will present a content-based image retrieval system which uses combination of low-level features for image retrieval as compared to previous retrieval schemes which only works over a single feature for content-based retrieval [1], [2].

Our research work targets to bring the MUVIS beyond the desktop environment into a realm of wireless devices (mobile phones, Personal Digital Assistants (PDAs), communicators etc). Our main goal is to design and develop a content-based image retrieval system, which would enable any client supporting Java [13] platform to retrieve images similar to query image from an image database. We have developed a system called Mobile MUVIS (M-MUVIS) based on contemporary MUVIS. M-MUVIS has a client-server computing architecture. The M-MUVIS server comprises of a Java servlet [14] running inside a Tomcat [17] web server which in effect transforms the standalone MUVIS into a web application. Servlets are Java programs, which act like add-on component to Java enabled server (e.g., Tomcat) and enhance the capabilities of it. The M-MUVIS server has native libraries for efficient query related operations of an image. To take advantage of flexibility and portability of Java, the M-MUVIS client has been developed using Java 2 Platform, Micro Edition (J2ME) [19]. It is oriented towards consumer devices possessing limited capabilities and functionalities as mentioned earlier. Such a system can find its application in sharing or reuse of digital media, content management, networked photo album, shopping and travel.

This paper is organized as follows: Section 2 gives an overview of MUVIS, which is the basis of the proposed M-MUVIS. Section 3 describes the architecture of M-MUVIS and Section 4 is dedicated to cover several functionalities of M-MUVIS. Section 5 contains experiments carried out over M-MUVIS. Finally in the last section we draw conclusions and provide hints for the future research plans.

2. MUVIS SYSTEM OVERVIEW

Content-based image retrieval (CBIR) addresses the problem of accessing the images that bears some certain content and usually relies on the characterization of primitive features such as color, shape and texture, all of which can be extracted from the images [3]. One of the CBIR systems is MUVIS, which aims to bring a unified and global approach on indexing, browsing and querying of various digital multimedia types such as audio/video clips and digital images. In order to achieve such a global objective MUVIS consists of a set of applications. The MUVIS application for querying, searching and browsing is called *MBrowser*. It has functionalities of an advanced multimedia player and built-in search and query engine. The underlying design of M-MUVIS uses the querying capability of *MBrowser*. Another MUVIS application that is also used in M-MUVIS is *DbsEditor*. It is designed for indexing multimedia databases and therefore, it is primarily used for feature extraction operations. The creation of image databases within MUVIS is elaborated in next sub-section.

2.1 Image Databases in MUVIS

DbsEditor is used for creating an image database and extracting/appending low-level features from the image database. The feature extraction is an offline process, which extracts low-level features based on shape, color and texture. Currently, the following features are available under MUVIS framework.

- RGB, HSV and YUV color histogram features.
- Dominant color based features.

- Gray Level Co-occurrence Matrix for texture features [6].
- Gabor Wavelet Transform method for texture features [5].
- Shape features extended from Canny Edge sub-sequent analysis.

Generally speaking, in order to create an image database, a dummy image database is first created and then images are appended to it. Once the data collection phase is completed (i.e. all images are appended), then the feature extraction can be performed over the database in order to perform retrievals via query operations. The overall operation with the supported image types is illustrated in Figure 1. Once all the operation is completed, the database can be represented by a file, which carries parametric information such as total number of images, feature descriptors, version information, etc.

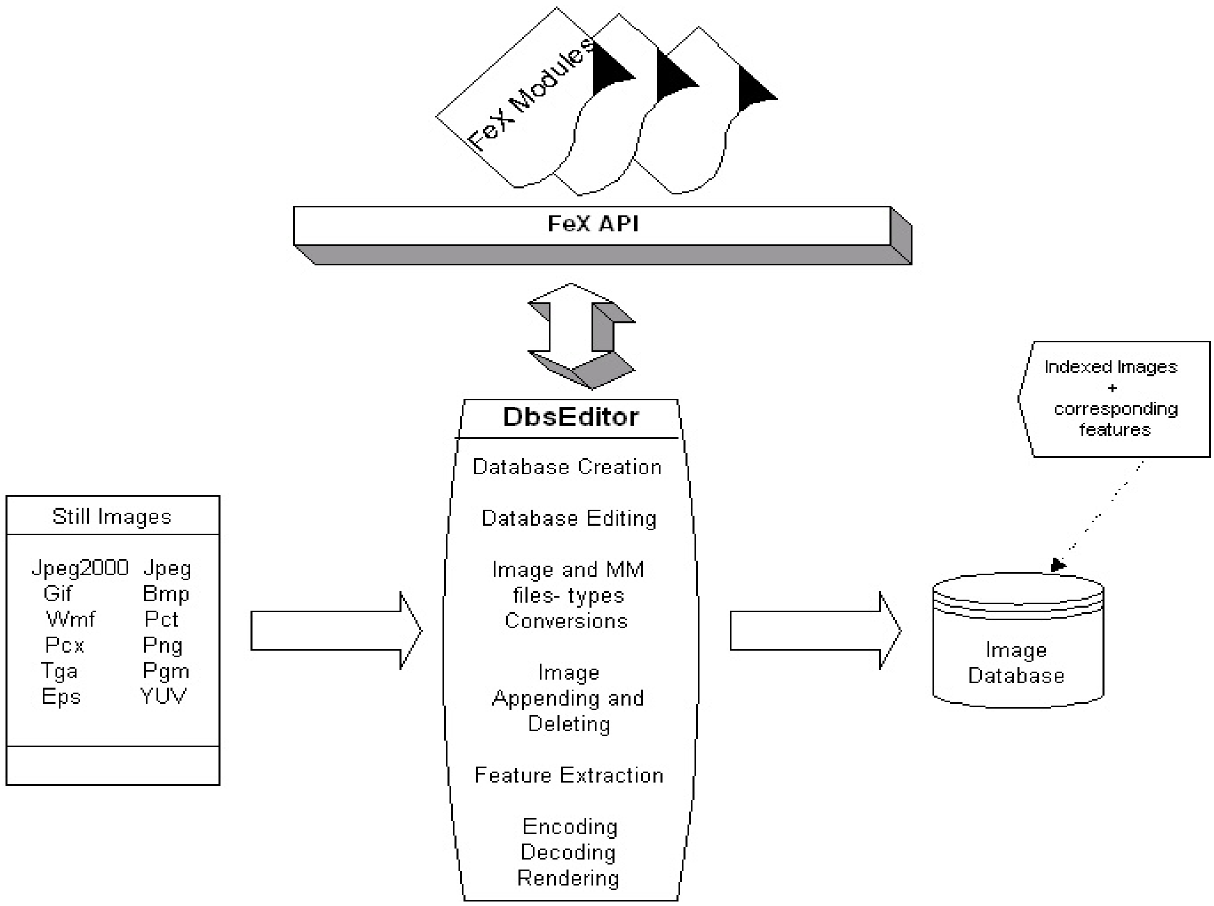

Figure 1. *DbsEditor,* A MUVIS application for database handling

3. ARCHITECTURE OF THE PROPOSED M-MUVIS SYSTEM

As mentioned earlier, M-MUVIS consists of client and server applications. The M-MUVIS server has a blend of Java and native C++ code so that we are making use of scalability, robustness and portability of Java and efficient implementation of native code. The communication protocol has been designed according to the requirements of information retrieval across a wide range of wireless devices. This architecture forms a basis of information retrieval on mobile devices.

In the following sub-sections we shall discuss the architectures of the client and server in detail.

3.1 The Client Architecture

The client application consists of three components/packages.

a) **Query Information Engine**: All the necessary query related information is encoded by this module and transferred to the communication protocol. The user can initiate some commands using User Interface (UI) of M-MUVIS client application. The commands are then processed and the corresponding actions are taken by this component.

b) **User Interface**: This component is responsible for displaying of query results and presenting a menu to the user for interaction. Using the menu option, the user can start a query operation. The user can view images in the query operation back and forth. Since the devices supporting Java MIDP [19] usually have a smaller screen size and therefore capable of showing one image at a time, the user can view the images one by one, using an option in the menu.

c) **Communication**: A communication protocol between M-MUVIS client and server has been defined. This protocol uses HTTP [11] as an underlying protocol. It transports the query request from M-MUVIS client over the General Packet Radio Services (GPRS) [15] network and brings the query response upon completion of query operation in M-MUVIS server. The protocol meets the requirements of content-based information retrieval. It has been kept simple, yet powerful enough to be used across wide range of embedded devices.

Figure 2 shows the general architecture of M-MUVIS and presents the main components of the client and server side.

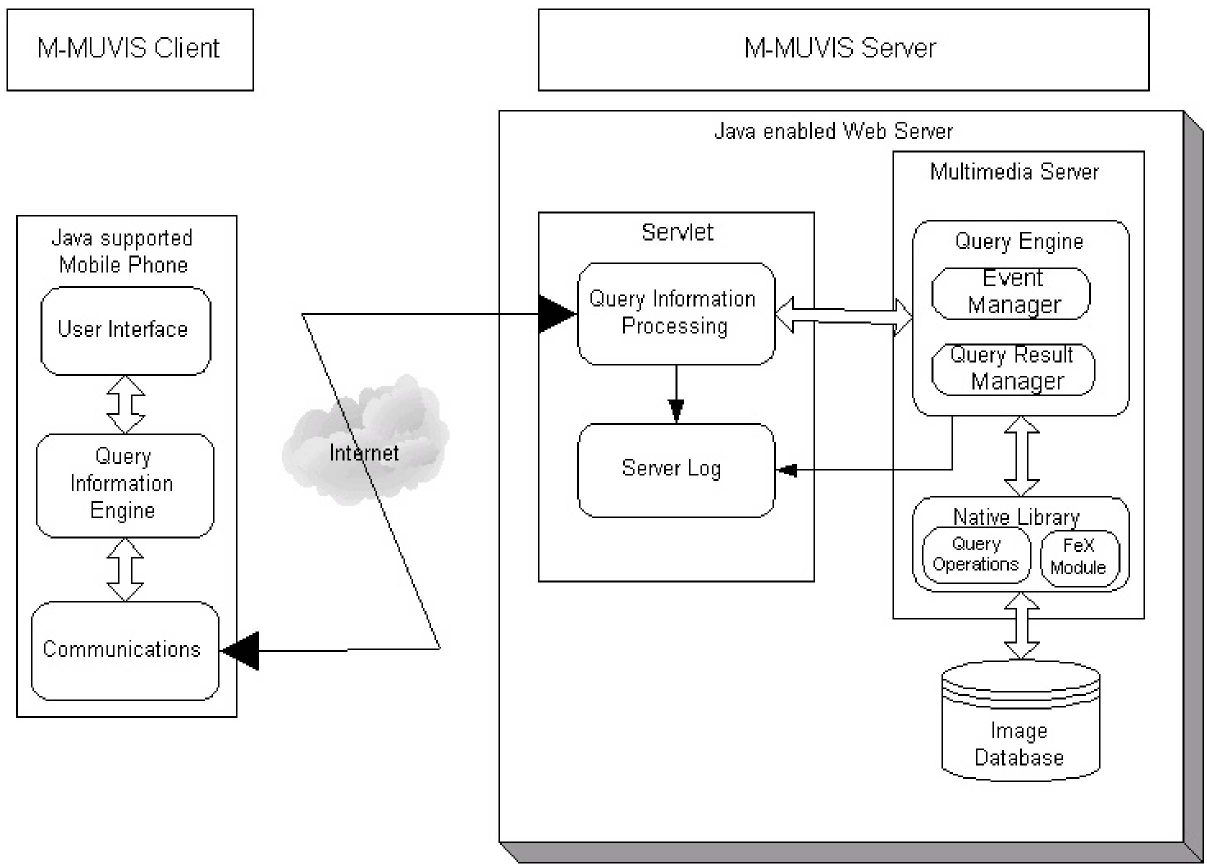

Figure 2. Architecture of M-MUVIS and its components

3.2 The Server Architecture

The main components of the M-MUVIS server are as follows:

- **Native Library**: The core query operations are implemented in *Native Library*. The *Native Library* has been created as a dynamically linked library (DLL) [18] and consists of libraries of native functions for carrying out processor intensive query operation more efficiently than the Java code in M-MUVIS serverThey include following operations:
  1. Loading of available database features from disc to system memory.
  2. Calculating similarity distances.
  3. Performing ranking operation with respect to the similarity distances.

  Feature eXtraction (*FeX*) module(s) is used to calculate similarity distances.

- **Query Information Processing**: It is responsible for parsing a query request and passing it to *Query Engine*. It then encodes the retrieval results obtained from *Query Engine* into a formatted string for fetching back to client
- **Server Log**: It records log of server operation related to query. It also contains information about the clients sending query request.
- **Query Engine**: *Query Engine* receives query request information from *Query Information Processing* module and performs query operation within *Native Library*. *Query Engine* acts as a layer of abstraction between *Native Library* and *Query Information Processing* module. It provides an interface for performing query operations in native (C++) code through Java Native Interface (JNI) [12]. It consists of following two components:
  - **Event Manager**: This component in *Query Engine* registers *Event Listeners* (a class, which implements *Event Listener Interface,* which an interface containing set of methods) and dispatches results to corresponding event listeners.
  - **Query Result Manager**: It is responsible for obtaining results from native portion in *Native Library* and passing them to *Query Information* module.

As explained earlier, the M-MUVIS server consists of a servlet running inside Java enabled Tomcat web server. It has a repository of images with sizes scaled down to dimensions suitable for display on mobile platforms. Upon receiving the query request from a client, the M-MUVIS server maps the request to the servlet. The query request is parsed and information is extracted by *Query Information Processing* module as shown in *Servlet* portion of Figure 2.*Query Information Processing* module passes query information to *Query Engine*, which performs query operation using *Native Library.* The query operation can use one or more combination of low-level features as mentioned in Table 1. The similarity distances are calculated by using the feature vector(s) of queried image with feature vector(s) of the images in a database. Ranking operation is performed afterwards and the retrieval result is formed using the best-12 ranked images. *Event Manager* in *Query Engine* notifies about the retrieval result event and *Query Result Manager* in *Query Engine* receives the retrieval results and sends them to *Query Information Processing* module*.* Query response is now in the form of a formatted string and encapsulated inside HTTP by *Query Information Processing* module for passing to client. The client upon receiving query response retrieves images one by one from M-MUVIS server. The images are displayed on mobile phone by M-MUVIS client application. The overall query process is illustrated in Figure 3.

3.3 Query Process in M-MUVIS

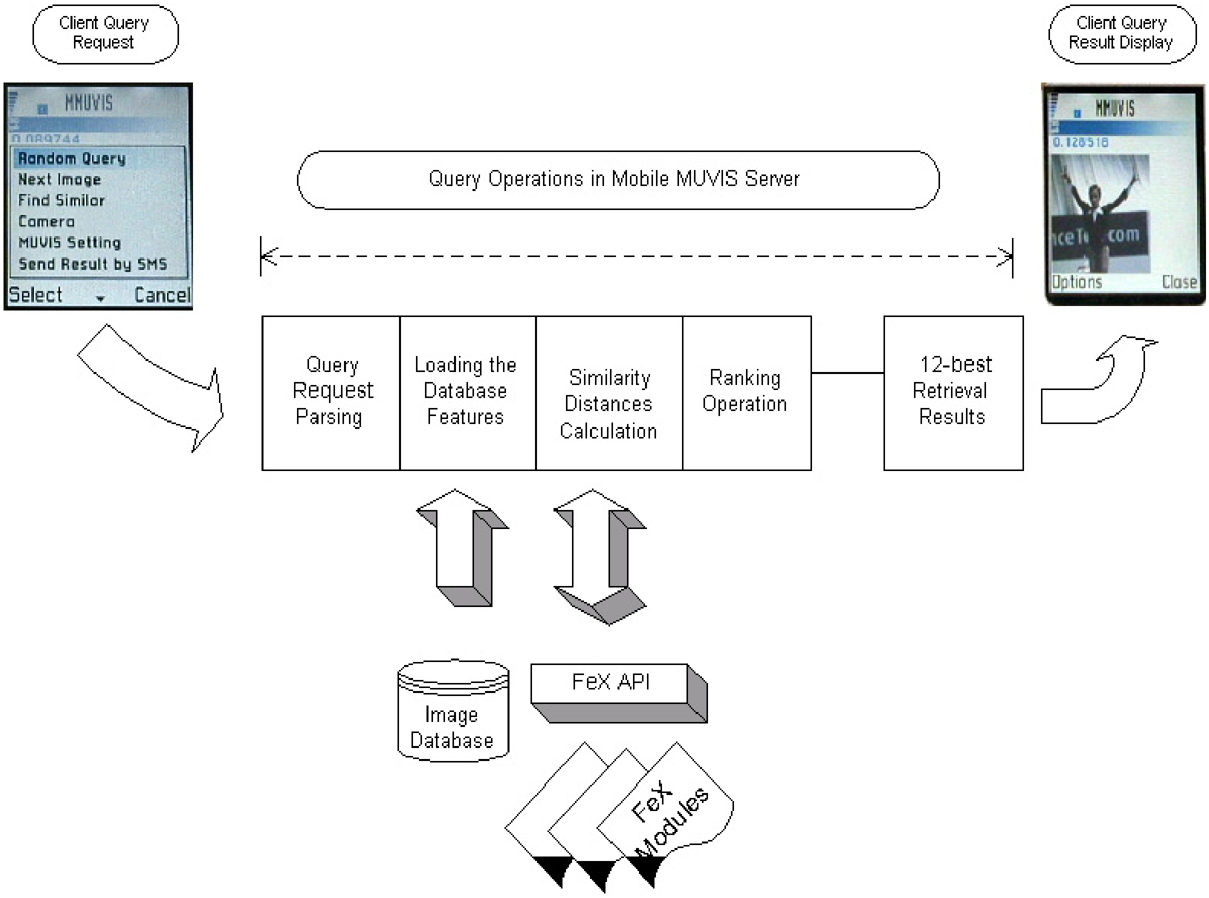

Figure 3. Query Process in M-MUVIS

4. M-MUVIS FUNCTIONALITIES

M-MUVIS client application basically provides the following functionalities to mobile phone user.

- Image Capture:

  Using this option in the menu of M-MUVIS client application the user can capture an image and later send it to the M-MUVIS server for a query operation.

- Random Query:

  In this type of query, client can perform a query operation using randomly selected images within image database of M-MUVIS server.

- Find Similar:

  This menu option is used to do a query operation on the retrieval results obtained from the last query operation. In this the client can get more similar images from image of interest in query result.

- Next Image:

  This option is used to retrieve next retrieval result of the last query operation. This option allows the user to retrieve and view the images one by one from M-MUVIS server.

- Send Result by SMS:

  This option is used to send the retrieval results to other mobile phone clients. By this way the same query result can be shared among several clients.

- Show Log:

  Every operation on M-MUVIS client side is recorded in a Log. The client can view the log containing information about query operations such as query request, image retrieval time, and size of images retrieved in sequential order of the operation.

5. EXPERIMENTAL RESULTS

We carried out a number of experiments for the performance evaluation of the proposed M-MUVIS. The image database we used in our experiments consists of around 2000 images with dimensions scaled down to suit display-size of a mobile screen. The specifications used in our image database are tabulated in Table 1. During our experiments the M-MUVIS server was running on a PC equipped with Intel Pentium 4 processor with 768 MB RAM (Random Access Memory) and running Microsoft Windows XP operating system.

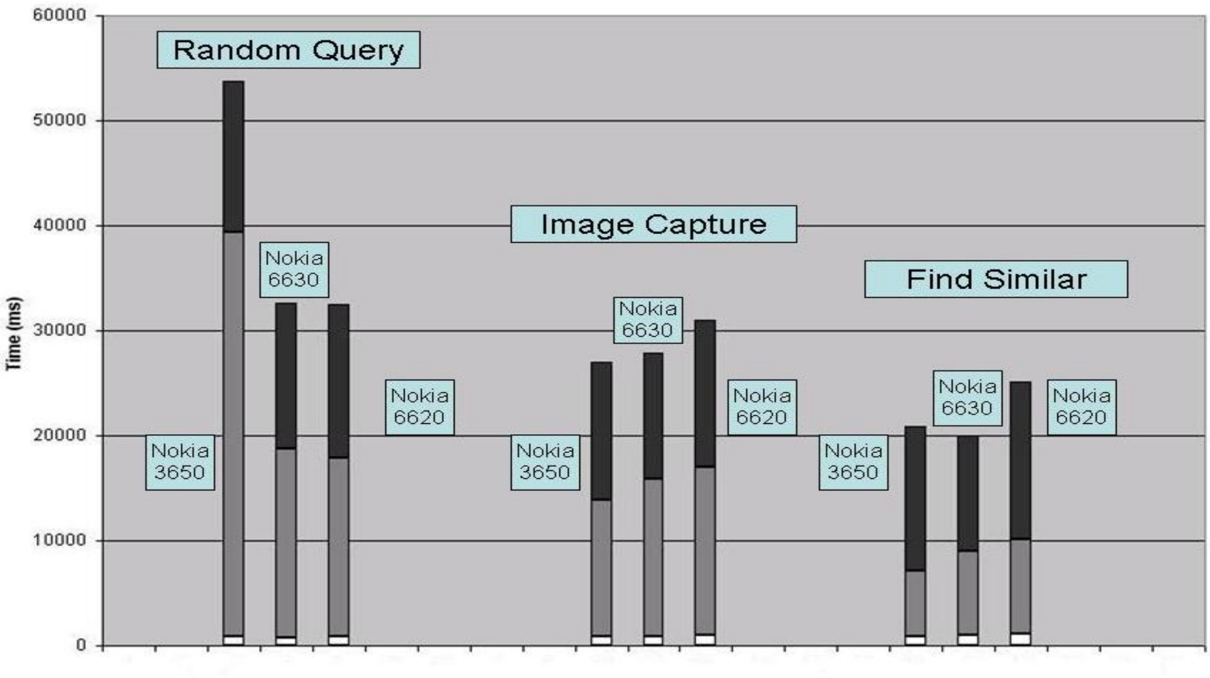

Figure 4. Query Timing Results on M-MUVIS

Following functionalities are used in our experiments: *Random Query*, *Find Similar* and *Image Capture* performed on different mobile hand-sets. Each bar in graph of Figure 4 is a sum of three timing results. These results represent following timing results in order from bottom to top:

- Query process time on M-MUVIS server is called *Server Query Time* shown by white region in a bar.
- Total query process time on M-MUVIS client is called *Client Query Time* shown by gray region in a bar.
- Time to receive first image among retrieved result is called *Image Retrieval Time* indicated by black region in a bar.

Server Query Time starts whenever a query request arrives to the servlet from a client. The servlet extracts image features and performs the query using native code. It includes the time to create a query result in a formatted string for sending it to the client. *Client Query Time* is the total cumulative time of sending the query request and receiving retrieval results from the server over transport channel whereas transport channel consists of mobile networks connected to Internet. It also includes *Server Query Time*. Finally, the *Image Retrieval Time* is the time to send the request of image downloading from retrieved results by the client plus the time to receive the actual image from the server file system.

Table 1. Image Database Specifications in M-MUVIS

Database Type	**No. of Images**	**Format**	**Dimensions**	**Features**
Image	2000	Png	Horizontal (70-176) pixels Vertical (100-200)	• HSV, and YUV color histogram features • Gray Level Co-occurrence Matrix for texture features

According to the results of Figure 4 major portion of time in query operation is taken up by the transport channel between client and server. Experimental results in Figure 4 show that only fraction of total query operation time (5-10%) is taken by server for processing the incoming query request and generating results. The unpredictable latency in transport channel of packet based mobile network (e.g., GPRS) as indicated in Client Query Time adds to high total query time experienced by the client. The Image Retrieval Time is in direct proportion to the size (in bytes) of image and may suffer delay due to a possible congestion of the transport channel during the query operation.

It can be concluded from query timing results that if the size of the database (number of images contained in it) increases approximately 20 times the size of database used in experiments as mentioned in Table 1, Server Query Time will then become comparable to Image Retrieval Time.

Figure 5.'Random Query' operation in M-MUVIS client and its retrieval results

Figure 6. 'Find Similar' query operation in M-MUVIS client and its retrieval results

6. CONCLUSIONS AND FUTURE WORK

We have designed a system M-MUVIS for content-based image retrieval on mobile devices. This system is an extension of contemporary MUVIS into a realm of embedded devices. Since the existing information delivery structure of World Wide Web is ill-suited for content distribution and reusing the information for diverse range of platforms including mobile phones and PDAs so we have proposed a system which has the capability to cope up with information management across heterogeneous networks and wide range of devices. The proposed system has been tested with an image database. We have obtained quite encouraging results based on content-based retrieval. However, we are facing limiting factors such as hardware/software features in mobile devices and unpredictable network responses. With 3G around the corner and available high-end devices such limitations and drawbacks can be overcome. M-MUVIS holds a promising future in information content management and retrieval as it can be tuned with different combinations and weights of low-level features to meet requirements of certain enterprise or private usage.

Currently, M-MUVIS Server is only a CBIR server and we are planning to make it a fully-functioning content-based multimedia indexing retrieval server. We aim to make M-MUVIS client application more user interactive, so that the user in our future work will be able to select the features upon which image retrieval is intended.

REFERENCES

[1] A. Iftikhar, F. Alaya Cheikh, B. Cramariuc and M. Gabbouj, "Query by Image Content using NOKIA 9210 Communicator" , Proc. of the Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'01, pp.133-137, Tampere, Finland, May 2001.

[2] A. Iftikhar, F. Alaya Cheikh, B. Cramariuc, and M. Gabbouj, "Query by Image Content Using Mobile Information Device Profile (MIDP)",Proc. of the 2003 Finnish Signal Processing Symposium, Finsig'03, Tampere, Finland, May 2003.

[3] S. Kiranyaz, K. Caglar, E. Guldogan, O. Guldogan, and M. Gabbouj. "MUVIS: a content-based multimedia indexing and retrieval framework", Proceedings of the Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003, Paris, France, pp. 1-8, July 2003.

[4] A. Kushki, P. Androutsos, K.N. Plataniotis, A.N. Venetsanopoulos, “Retrieval From Artistic Repositories Using a Decision Fusion Framework”, IEEE Transactions on Image Processing, pp. 277-292.vol. 13, 2004.
[5] W. Y. Ma, B. Manjunath,”Texture Features for Browsing and Retrieval of Image Data”, IEEE Trans. PAMI, vol. 18, pp 837-842, Aug. 1996.
[6] M. Partio, B. Cramariuc, M. Gabbouj, and A. Visa, “Rock Texture Retrieval using Gray Level Co-occurrence Matrix”, NORSIG-2002, 5th Nordic Signal Processing Symposium, On Board Hurtigruten M/S Trollfjord, Norway, October 2002.
[7] A. Pentland, R.W. Picard, S. Sclaroff, “Photobook: tools for content based manipulation of image databases”, Proc SPIE (Storage and Retrieval for Image and Video Databases II) 2185:34-37, 1994.
[8] R. Sarvas, E.Herrarte, A.Wilhelm, and M.Davis, “Metadata Creation System for Mobile Images”, Proc. of the 2nd international conference on Mobile systems, applications, and services, MobiSys, Boston USA, Pages: 36 -48, June 2004.
[9] J.R. Smith and Chang, “Visual SEEk: A fully automated content-based image query system”, ACM Multimedia, Boston, Nov. 1996.
[10] K.Tollmar, T.Yeh and T.Darrell, “IDeixis: image-based Deixis for finding location-based information”, Mobile HCI, Vienna, Austria, Pages: 781 – 782, 2004.
[11] HTTP Pocket Reference (Pocket Reference (O'Reilly)) by Clinton Wong.
[12] Java (TM) Native Interface: Programmer's Guide and Specification by Sheng Liang.
[13] Java How to Program, 5th Edition by H. M. Deitel, P. J. Deitel, Harvey M. Deitel, Paul J. Deitel.
[14] Java Servlet Programming (O’Reilly) by Jason Hunter with Willaim Crawford.
[15] J. Lempiäinen, M. Manninen, Radio Interface System Planning for GSM/GPRS/UMTS, Kluwer Academic Publishers, 2001.
[16] “MUVIS”, http://muvis.cs.tut.fi.
[17] Professional Apache Tomcat 5 by Vivek Chopra, Amit Bakore, Jon Eaves, Ben Galbraith, Sing Li, Chanoch Wiggers.
[18] Programming Windows, Fifth Edition by Charles Petzold.
[19] The Complete Reference J2ME by James Keogh, McGrawHill OSBORNE Edition.
[20] “3rd Generation Partnership Project (3GPP)”, http://www.3gpp.org/About/about.htm.
[21] “Virage”. http://www.virage.com.