

# An Efficient Image Retrieval Scheme on Java Enabled Mobile Devices

Iftikhar Ahmad  
Nokia Corporation  
P.O.Box 88, FIN 33721  
Tampere, Finland  
Email: [iftikhar.ahmad@nokia.com](mailto:iftikhar.ahmad@nokia.com)

Serkan Kiranyaz, Moncef Gabbouj  
Signal Processing Laboratory  
Tampere University of Technology, P.O.Box 553  
FIN-33101, Tampere, Finland  
Email: [serkan@cs.tut.fi](mailto:serkan@cs.tut.fi), [moncef.gabbouj@tut.fi](mailto:moncef.gabbouj@tut.fi)  
<http://muvis.cs.tut.fi>

**Abstract**— Image retrieval over wireless networks is a challenging research problem. To achieve acceptable system performance one needs to solve several challenging problems related to the network and media characteristics, system usability and users expectations. The user expects to receive the retrieval results in a few seconds after putting his query. In this paper we focused on reducing the query and media retrieval time. The implementation of an advanced retrieval scheme is presented. This scheme is called Compact Media Retrieval on Progressive Query (CMRPQ) on mobile platforms. The CMRPQ is designed to reduce the media retrieval time and bandwidth usage, which are the two most critical bottlenecks of this application. The experimental results prove that the query and media retrieval time over different mobile devices in different networks can be reduced significantly.

**Keywords**— Content; J2ME; retrieval; image; mobile.

## I. INTRODUCTION

The mobile phone industry is going through a phenomenal change over the past few years with significant advances in the areas of communications and multimedia. 3G [5] 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. Nowadays, mobile platforms support Java [10] that provides rich programming APIs (Application Programming Interface). 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 media archives. Therefore, it has become vital to retrieve desired information expeditiously and efficiently using these devices.

Content-based image retrieval (CBIR) addresses the problem of accessing the images that bears some certain content and usually relies on the characterization of low-level features such as color, shape and texture, all of which can be extracted from the images. CBIR 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 for the past decade. The content-based retrieval of a desired multimedia item is currently based upon indexing of the content by the extraction of low-level visual features based on shape, color and texture.

Systems such as “Multimedia Video Indexing and Retrieval System” (*MUVIS*), [12], VisualSEEk [7], Photobook [6] and Virage [13] have a framework designed for indexing and retrieving images and/or audio-video files. The contemporary *MUVIS* has been developed as a system for content-based multimedia retrieval on a PC-based environment. It provides a unified and global framework that consists of robust set of applications for capturing, recording, indexing and retrieval combined with browsing and various other audiovisual and semantic capabilities.

With the encouraging results of content-based image retrieval [1], [2], [4], perform over mobile platforms and due limitations imposed by text-based queries researchers and scientists have undertaken challenge of meeting content management of users using mobile devices. In this paper, we present a CBIR system, which uses a combination of various low-level features. Along with the modified approach presented in this paper, the proposed scheme thus offers a significant improvement as compared to previous retrieval schemes, [1], [2], which only work over a single feature for content-based retrieval.

On this purpose our research work targets to bring the *MUVIS* framework beyond the desktop environment into the realm of wireless devices such as mobile phones, Personal Digital Assistants (PDAs), communicators etc., where the user can perform query operations in large multimedia databases and query results can be retrieved within a reasonable time. Therefore, our main goal is to design and develop a CBIR system that enables any (mobile) client supporting Java platform to retrieve images similar to the query image from an image database, which is accompanied by a dedicated server application. The developed system, so called Mobile *MUVIS* (*M-MUVIS*), is structured upon contemporary *MUVIS* framework and has client-server architecture. The *M-MUVIS* server basically comprises of two Java servlets [9] running inside a Tomcat [8] web server, which in effect transforms the

standalone *MUVIS* into a web application. The *MUVIS* Query Server (*MQS*) has native libraries for efficient image query related operations. The second servlet so called *MUVIS* Media Retrieval Server (*MMRS*) is used for the media retrieval. In order to take the advantage of flexibility and portability of Java, a *M-MUVIS* client application has been developed by using Java 2, Micro Edition (J2ME) [11]. Such a system can find its application in sharing or reuse of digital media,

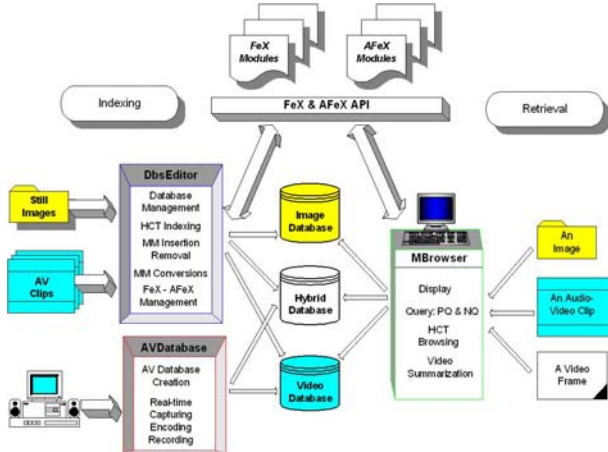


Figure 1: MUVIS framework

content management, networked photo album, shopping and travel.

This paper is organized as follows: Section 2 gives a brief overview of the *MUVIS* framework, which is the basis of the proposed *M-MUVIS*. Section 3 describes the basic architecture and several functionalities of *M-MUVIS*. Section 4 present several experiments carried out over *M-MUVIS*. Finally we draw conclusions in section 5.

Table 1: MUVIS supported Image Types

Convertible Formats							
JPEG	JPEG 2K	BMP	TIFF	PNG			
Non-convertible Formats							
PCX	GIF	PCT	TGA	PCX	EPS	WMF	PGM

## II. MUVIS SYSTEM OVERVIEW

*MUVIS* is a system, 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 that are illustrated in Figure 1. *DbsEditor* and *AVDatabase* are the main applications used for database creation and organization. *MBrowser* application is used for browsing, query operations summarization and also has the generic functionalities of an advanced multimedia player/viewer.

The underlying design of *M-MUVIS* uses the querying capability of *MBrowser* and the entire *DbsEditor* application

for offline indexing and feature extraction operations performed over multimedia databases. *MUVIS* supports a wide range of multimedia formats; a list of supported image formats is given in Table 1.

### A. Progressive Query

The most common retrieval scheme is query-by-example (*QBE*) the traditional query operation, which is usually performed via an exhaustive search over the entire database due to lack of an efficient indexing scheme is called as Normal Query (*NQ*) [4]. Such a basic *QBE* operation works as follows: using the low level features of the query image and the database images, the similarity distances are first calculated per feature basis and then merged to obtain a unique similarity distance per database item. Ranking according to the similarity distances over the entire database yields the query result. Progressive Query (*PQ*) [3] can provide faster query results along with the query process and let the user to browse the intermediate results. *PQ* is composed of a periodic series of Progressive Sub-Queries (*PSQs*). A sub-query, which is a partial query operation performed over a sub-set of database items, is based on a specific time period that can be a user preference before initiating the query operation. The database items used in a sub-query can be selected either randomly or sequentially. In our experiments it is selected sequentially.

The first *PSQ* is the first periodic sub-query performed. After the first *PSQ*, the rest of the *PSQs* are obtained by a fusion operation between the current periodic sub-query and the previous *PSQ*. The fusion operation is a process of fusing two of the sorted sub-query results to achieve one (fused) sub-query result. Since both sub-query results are already sorted with respect to the similarity distances of the items within the

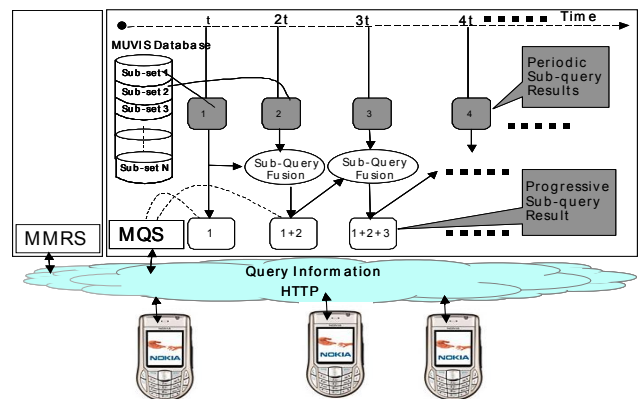


Figure 2: M-MUVIS framework

sub-sets, simply comparing the consecutive items in each of the sub-query lists can perform fusion. More information about *PQ* can be obtained in [3]. *PQ* operation in *MQS* is illustrated in Figure 2.

## III. M-MUVIS SYSTEM ARCHITECTURE

As mention earlier, *M-MUVIS* is a client-server framework, where servers are running on a computer and

client application is running on mobile devices (Nokia’s 6630, 6600 [15] etc). A particular *M-MUVIS* client can send a query request to *MQS* and *MQS* then responds by sending the query results to the client. For the retrieval of the query resultant image, *M-MUVIS* client sends the request to the *MMRS*.

#### A. *M-MUVIS* client

*M-MUVIS* client consist of three parts: User Interface (*UI*), *Engine* and *Utility*. The user can initiate several operations such as query, change of *M-MUVIS* client settings, etc. and *Engine* can use the *UI* to display the query resultant image or arbitrary images in varying resolutions.

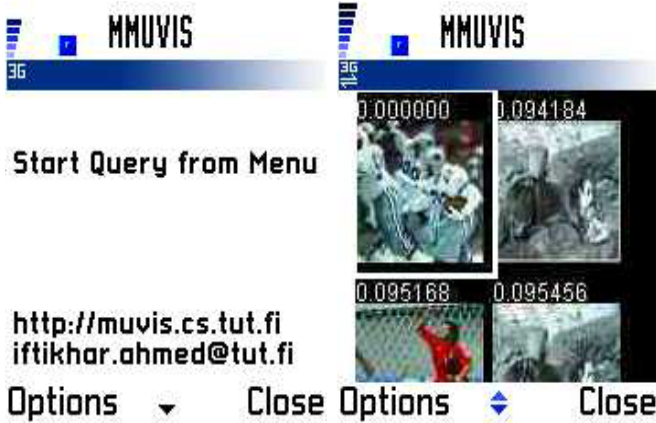


Figure 3: *M-MUVIS* client UI with a resultant image.

*UI* display capability can support different screens and image sizes. During the startup phase of *M-MUVIS* client, *UI* is adapted according to device screen size. The same *M-MUVIS* client application can thus be used on different devices with different screen sizes. Main view of *M-MUVIS* client and a query resultant image (*QRI*) as compact media is shown in Figure 3. In the rest of the paper compact media is refer as *QRI*.

The *Engine* part is responsible for the activities behind the scene. It determines the thumbnail size according to the screen size of the device. The user may then change such default setting of the thumbnail image size from the settings dialog. When the user initiates a query, the engine performs the query operation by contacting the *MQS* and retrieves the query results as an encoded string from the *MQS*. After retrieving the query results from *MQS* as encoded string, it sends a request to *MMRS*, to retrieve the *QRI*. The *Engine* is also responsible to maintain the list of thumbnails. It can later be used to retrieve a particular image or to initiate another query using the menu.

The *Utility* is the third part, which consists of commonly used Java classes for string handling used in *Engine* and *UI*.

#### B. *M-MUVIS* query server

*MQS* is responsible for performing the query operation on the server side. It uses native libraries (C/C++ code) to perform a query operation. When *MQS* receives the query request from an *M-MUVIS* client, it first parses the query string and then passes the query request to the native side. Upon completion of the requested *PQ* operation, the native

side sends the query results to the *MQS*. Then *MQS* sends the requested *PQ* result to the *M-MUVIS* client.

#### C. *MUVIS* media retrieval server

*MMRS* creates the query resultant image where the first twelve similar images are drawn as thumbnails in a so called “query resultant image”. Similarity score (distance) of each image is drawn on top of it as shown in Figure 4. Later *MMRS* (JPEG) compresses the *QRI* and send it to the *M-MUVIS* client. *MMRS* can also provide images to *M-MUVIS* client on its request.



Figure 4: a query resultant image shown in a Nokia 9500 mobile phone screen

#### D. Protocol between client and servers

In order to specify the internal settings of a particular query operation, the *M-MUVIS* client, *MQS* and *MMRS* use stream messages in an encoded string format.

The encoded string between client and server uses HTTP [14] protocol underneath. HTTP is a stateless protocol, so a session is created in *MQS* when the query request is received from an *M-MUVIS* client as shown in Figure 2. Session tracking allows the *M-MUVIS* client to retrieve an intermediate *PSQ* result. The specified *PSQ* result is then transmitted to *M-MUVIS* client.

Table 2: Features in image database

Database type	Features
Images	<ul style="list-style-type: none"> <li>• HSV and YUV color histogram</li> <li>• Gray Level co-occurrence matrix for texture</li> </ul>

## IV. EXPERIMENTAL RESULTS

The image database used in the experiments is consisted of 1867 images in PNG (Portable Network Graphics) format. The visual features extracted for the image database are tabulated in Table 2. During our experiments the *MQS* and *MMRS* were running on a PC equipped with Intel Pentium 4 processor 2.13GHz, 1.50 GB RAM (Random Access

Memory) and running Microsoft Windows XP operating system.

Client Query Time (*CQT*) is the waiting period for the query results measured on an *M-MUVIS* client. An

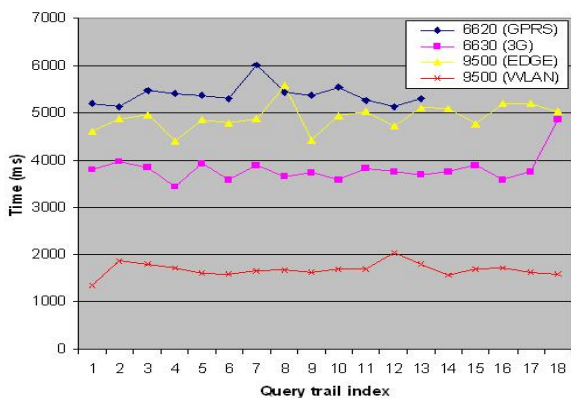


Figure 5: Client query time (CQT)

experimental *CQT* in milliseconds measured on Nokia 9500, 6620 and 6630 is plotted in Figure 5 along y-axis. We measure the *CQT* using different networks such as WLAN (Wireless Local Area Network), 3G, EDGE, GPRS. The fastest query time is so far achieved in WLAN whereas *EDGE* performs the query operation lasting in between *3G* and *GPRS*.

Client Media Retrieval Time (*CMRT*) is the time on the *M-MUVIS* client for the resultant image retrieval. An experimental *CMRT* is measured on Nokia 9500, 6620 and 6630 as plotted in Figure 6 where *CMRT* in milliseconds is shown along y-axis and resultant image quality factor of JPG is shown along x-axis. As shown in the Figure 5 and Figure 6, *WLAN* has minimum network latencies so the fastest query and media retrieval operations can be achieved over *WLAN*. Due to dynamic nature of wireless networks graphs shown in Figure 5 and Figure 6 are not linear.

Since in practice the mobile devices cannot support high-resolution display, high compression rates can be conveniently

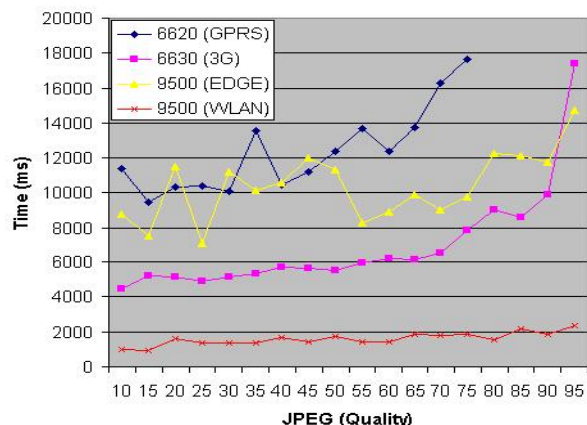


Figure 6: Client media retrieval time (CMRT)

used for the retrieval (transmission) of the images.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, an efficient compact media retrieval framework on mobile devices is proposed, in order to achieve the following innovative properties:

- *PQ* can be used as the primary QBE retrieval scheme on mobile platforms in order to obtain faster query results with the ongoing query process.
- As a result of using a compact media (submerged) resultant image to retrieve several images in reduced (thumbnail) dimensions, the media retrieval time is significantly reduced per *PSQ* retrieval.
- Furthermore, the formation of the query resultant image allows that more query results can be retrieved together so that the user can compare them easily.

We foresee that *PQ* can also be applicable on the retrieval via QBE over audio and video databases for mobile platforms where-as the protocol between client and server can be optimized for faster retrievals. Therefore, we will focus the efforts on the extension of the proposed scheme for a generic multimedia support within *M-MUVIS* framework.

## REFERENCES

- [1] Ahmad Iftikhar, Faouzi Alaya Cheikh, Bogdan Cramariuc and Moncef Gabbouj, Query by Image Content using Nokia 9210 Communicator, WIAMIS 2001 Workshop on Image Analysis for Multimedia Services 16 – 17 May 2001 Tampere, Finland.
- [2] Ahmad Iftikhar, Faouzi Alaya Cheikh, Bogdan Cramariuc and Moncef Gabbouj Query by image content using Mobile Information Device profile (MIDP), Finsig' 03, Tampere International Center for Signal Processing, Tampere, Finland, May 19, 2003.
- [3] S. Kiranyaz, M. Gabbouj, A Novel Multimedia Retrieval Technique: Progressive Query (WHY WAIT?), WIAMIS Workshop, Lisboa, Portugal, April 2004.
- [4] I. Ahmad, S. Abdullah, S.Kiranyaz, M.Gabbouj, "Content-Based Image Retrieval on Mobile Devices", Proc. SPIE (Multimedia on Mobile Devices) 5684, Electronic Imaging Symposium 2005, San Jose, California (USA), 16-20 Jan. 2005.
- [5] J. Lempiäinen, M. Manninen, Radio Interface System Planning for GSM/GPRS/UMTS, Kluwer Academic Publishers, 2001.
- [6] 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.
- [7] J.R. Smith and Chang, VisualSEEK: A fully automated content-based image query system, ACM Multimedia, Boston, Nov. 1996.
- [8] Professional Apache Tomcat 5 by Vivek Chopra, Amit Bakore, Jon Eaves, Ben Galbraith, Sing Li, Chanoch Wiggers.
- [9] Java Servlet Programming (O'Reilly) by Jason Hunter with Willaim Crawford.
- [10] Java How to Program, 5th Edition by H. M. Deitel, P. J. Deitel, Harvey M. Deitel, Paul J. Deitel.
- [11] The Complete Reference J2ME by James Keogh, McGrawHill OSBORNE Edition.
- [12] "MUVIS", <http://muvis.cs.tut.fi>
- [13] "Virage," <http://www.virage.com>
- [14] HTTP Pocket Reference (Pocket Reference (O'Reilly)) by Clinton Wong.
- [15] <http://www.nokia.com>