Query by Image Content using NOKIA 9210 Communicator

Ahmad Iftikhar[1], Faouzi Alaya Cheikh[2], Bogdan Cramariuc[2] and Moncef Gabbouj[2]

Abstract— In this paper we present a new Java-based client-server application for content-based image retrieval over wireless networks. The application on the client side is running on the NOKIA's 9210 Communicator and is written in pure Java™.

Index Terms— Content, Retrieval, Indexing, Multimedia, Image, Search, Wireless, Mobile, Communicator.

I. INTRODUCTION

A. Wireless Communications and Terminals

The way people are communicating is changing very fast. Few years ago, mobile phones were lucrative items restricted to a very small community of rich businessman and government agents. Moreover, they were used exclusively for voice calls.

Today the mobile terminal penetration is growing steadily and continuously. And their use is no longer restricted to voice communication only. In Finland, it is widely accepted among youngsters to use a GSM phone for sending SMS messages, to chat with friends or to play games. Adults may be more interested in checking their stocks or paying a bill using their wireless terminal and the Wireless Application Protocol (WAP). In Japan a phenomenal change in the use of mobile phones happened by the introduction of the "iMode" [IMODE] system. The number of users since its introduction two years ago has risen to 17 millions.

The third generation, or 3G [3G], phones will create new opportunities for content providers, by providing a way of transmitting text, voice, images, and streamed video. Moreover, their ability to be connected to the Internet all the time will provide users with an overwhelming access to a huge amount of information. Users will then face the problem of how to retrieve the information of interest to them in an efficient manner. The goal is to allow for searching and navigation in this wealth of data without the need to make text-based queries for three obvious reasons:

- The user may be unable to type in commands.
- The keyboards of portable devices are not very comfortable for text-based commands.
- Text-based queries may not be very appropriate in the case of images, video or music.

Therefore, a content-based indexing and retrieval engine coupled to a speech recognition engine could be the ultimate interface to such a system

In this paper we will introduce a content-based search engine and its graphical user interface. A demo of the system will be given during the presentation. The speech recognition part is not considered in this paper.

Even though, the newly introduced pervasive devices are having faster processors, larger memories and their available communication bandwidth is getting wider, they remain far behind the PC capabilities. Therefore, a major challenge in designing such a system is to understand the characteristics of such devices and their hardware and software limitations.

B. Content-based Indexing and Retrieval

Since the early 1990s, content-based indexing and retrieval (CBIR) of digital images became a very active area of research. Both industrial and academic systems for image retrieval have been built. Most of these systems (e.g. *QBIC*™ [QBIC] from IBM, *NETRA* [NETRA] from UCSB, *Virage* [VIRAGE] from Virage Inc., *MUVIS* [MUVIS] from TUT) support one or more of the following options: browse, search by example, search based on a single or a combination of low level features. These features can be extracted from the image, such as color, shape, texture, spatial layout of objects in the scene or added to it after its capture, such as contextual information and keywords

II. CLIENT-SERVER ARCHITECTURE

A. The Client Side: The Nokia 9210 Communicator

1) Introduction

Nokia 9210 Communicator [NOKIA] is a major step forward in the road to the Mobile Internet environment. This pioneering product showcases the key elements in future mobile communications, such as easy navigation and input, a high-quality color display, mobile messaging with high data speed, imaging and video clips. Additionally, Java support and Symbian's OS (operating system) [EPOC]

[1] Nokia Mobile Phones, P.O. Box 1000 (Visiokatu 3, Tampere 33720), FIN-33721 Tampere, Finland

[2] Tampere University of Technology, P.O. Box 553, FIN-33101, Tampere, Finland

bring open development interfaces to the Nokia 9210 Communicator for numerous additional applications to be provided by any third party developers.

2) Hardware Details

Nokia 9210 communicator [9210F] contains 32-bit ARM-based RISC processor. It has 8 MB (SD-RAM) of execution memory and its C drive (serial flash) is of 4MB. It can have a multimedia card of up to 64MB, see Figure 1.

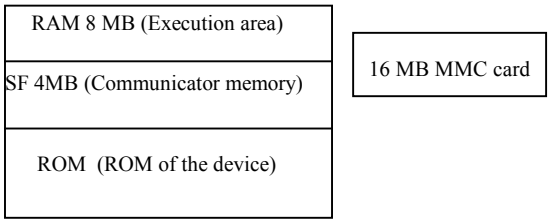

Figure 1: Memory configuration of the Nokia 9210

It has a color display of 4096 colors. Display size is 640 x 200 pixels. In addition, it has a relatively large size keyboard and is capable of making high speed data calls. It uses Symbian's operating system Crystal 6.0.

Java virtual machine consumes about 2.1MB. The proposed implementation of the Java application consumes approximately 397KB of memory.

3) Operating System

Symbian's platform [SYMB] is a robust, object oriented operating system for devices with limited capabilities (small memory, little computing power, sensitive to power consumption). Devices using this system do not need to reboot often as this OS is stable, does not leak memory (or very less) and manages the system resources efficiently.

Since Symbian's platform devices have little memory, small secondary storage and less computational power, applications written for this systems must be efficient. This is especially the case of Symbian Crystal release 6.0 [CRYST] intended for wireless media.

The proposed image search engine deals with images and thus consumes a large amount of memory. The system must thus be well managed to avoid such memory related problems. A high-speed data link is used. Actually the 9210 supports data links up to 43.2KB (High Speed Circuit Switched Data, HSCSD) [9210F], but we are using 38.4KB. High-speed data call reduces airtime but it is a costly option. Airtime will be reduced in 3G systems, and thus queries can be made without using a high-speed data call. Only a high-speed connection for data transfer will be needed.

Personal Java [PJAVA] is ported on the Crystal 6.0 that is compatible with JDK1.1.8 [JDK]. But current implementation of Personal Java is not supporting swing (a pure graphics APIs of Java). Personal Java consumes 2.1 MB of memory when just VM is up (without Java application). The Java Application (image search engine) takes an additional 397 KB of RAM leaving very little memory for images. In this implementation, images are fetched when requested to be displayed and discarded when not need.

B. The Server Side

On the server side we are using a servlet [SERV]. Servlets are Java programs that extend the capabilities of the server. They are similar to applets in a browser. The client sends the query to the servlet; which checks the query media type and passes it to the appropriate query handler.

The heavy processing required for the feature extraction, similarity estimation and results presentation are done on the server through calls from the Java side to methods implemented in native code. In this way we take advantage of the more efficient native code as compared to the pure Java implementation.

C. Communication Protocol

A communication protocol is defined between the client and the server. This Protocol specifies the media type (Image media, Video media, or Audio media, currently we are using Image media only), query type (random query from database, query by image data or query with an image from the database) and query data (image data if the image is not in the database or images' index in the database or image location URL).

The server sends back to the client the status of query execution and the results of the query, which consists of the list of names of the images and their similarity scores with respect to the query image. The client later fetches scaled versions (80 x 60 pixels) of the images to be presented to the user (in our experiments we requested 10 images). Scaling is done on the server side, in order to reduce the traffic. Only on the request of the user the full size image is fetched from the server.

III. THE USER INTERFACE AND SCREEN SIZE CONSIDERATION

As mentioned earlier, wireless devices have limited resources. In this application addition to the processing and memory issues, the designer has to consider the screen size of the wireless device. As the 9210 belongs to the communicator class, it has a relatively larger screen (640 x 200) [9210F]. When displaying the query results, images have been resized to fit the available display. As can be seen in the examples given in Figures 2-5, the image content can still be legible. Furthermore, in the 9210 we take advantage of the command button area and place the four most used commands there. The other commands are placed in the menu. The menu is displayed only when the user presses the menu button, and hence it is not consuming screen space when it is not active.

IV. Results and Assessments

Figure 2 shows the GUI and the menu toolbar on the Nokia 9210 Communicator which has been implemented for the proposed content-based image indexing and retrieval system. As can be seen, the screen space is fully utilized and the important features are displayed. Remember that the menu is not accessed very often, and thus the space is used to display the query results. The most commonly used buttons are assigned to the command button area on the right side of the screen.

Figures 3-5 show the results of different types of queries made to the image database, namely, color histogram, shape and texture queries. In each case, the top ten similar images are retrieved and displayed on the Communicator screen.

Due to the limitations imposed by the wireless device, the operating system and the communication channel, a rather slow query response has been achieved. Table 1 below shows the timing obtained in different queries made with the Nokia 9210 Communicator. Query time on Server starts when a query arrives at the servlet, the servlet extracts the image features and makes the query in the native code. It includes also the time to save the results on the server and creates a Java result object and passes it to the servlet to send it to the client. Sending result object to the client is the time to send Java object containing the image names and similarity scores for 50 images. The image retrieval time starts when the server passes the image retrieval request to the servlet. It includes the time needed to retrieve the actual images from the server's file system and resize them on the server side. Finally, the image transfer time is the time it takes the server to transmit the resized images to the client.

Query type	Shape	Histogram	Texture
Query time on server (sec)	26	17	16
Sending result object (ms)	100	110	108
Image retrieval (ms)	230	245	260
Image transfer time (ms)	430	470	490

Table 1: Timing Results for Image Query

The timing provided in Table 1 should be interpreted with care. They varied quite a lot during the testing as they depend on a number of rather dynamic factors, such as the network traffic (load), the load on the server (as well as the state of the server, i.e., servlet must be uploaded again in case no one requested its use from the server) and the available memory on the device.

V. Conclusions and Future Work

A novel implementation of TUT's MUVIS image query system has been proposed and tested on the new Nokia 9210 Communicator using a Java-based client server paradigm. A functional GUI was implemented taking into account the small size of the Communicator. The Demo shows that such an implementation is feasible; however, due to the limiting factors in both the hardware and software of the wireless terminal as well as the communication channel, very limited results have been obtained, namely, reduced sizes of image query results, small number of images, long process and access times. The good news is that with the advent of 3G networks, offering higher data rates and more processing power in wireless devices and more memory, such an application would be possible. Furthermore, a more efficient Java implementation, called J2ME (JAVA 2 Micro Edition) [J2ME], is under development. This implementation is targeted, among other applications, to small size wireless devices with limited capabilities.

VI. Acknowledgements

We gratefully acknowledge the support of Mr. Timo Ulmanen, from Nokia Mobile Phones for his efforts and support on behalf of this work.

VII. References

[IMODE] http://www.ntt.docomo.com/i/

[3G] http://www.3gpp.org/

[QBIC] http://wwwqbic.almaden.ibm.com/~qbic/

[NETRA] http://maya.ece.ucsb.edu/Netra/

[VIRAGE] http://www.virage.com/

[MUVIS] M.Trimeche, F.Alaya Cheikh, M.Gabbouj and Bogdan Cramariuc, "Content-based Description of Images for Retrieval in Large Databases:MUVIS," X European Signal Processing Conference, Eusipco-2000, Tampere, Finland, September 5-8, 2000, pp. 139-142.

[J2ME] http://java.sun.com/j2me/

[PJAVA] http://java.sun.com/products/personaljava/

[JDK] http://java.sun.com/products/jdk/1.1/

[SERV] http://java.sun.com/products/servlets/

[NOKIA] http://www.nokia.com/phones/9210/index.html

[9210F] http://www.nokia.com/phones/9210/features.html

[SYMB] http://www.symbian.com/

[CRYST] http://www.symbian.com/technology/v6-papers/v6-papers.html

[EPOC] http://www.epocworld.com

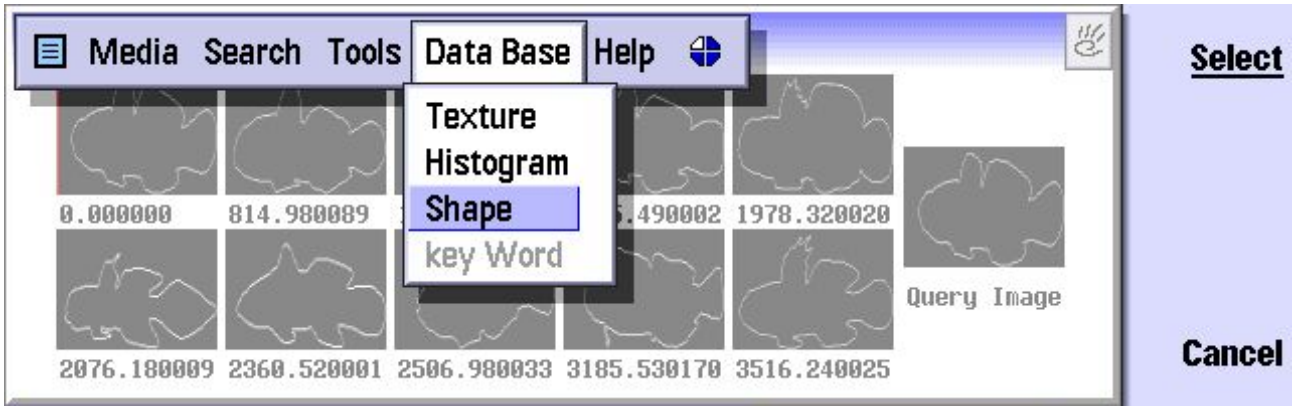

Figure 2: GUI on the Nokia 9210 Communicator

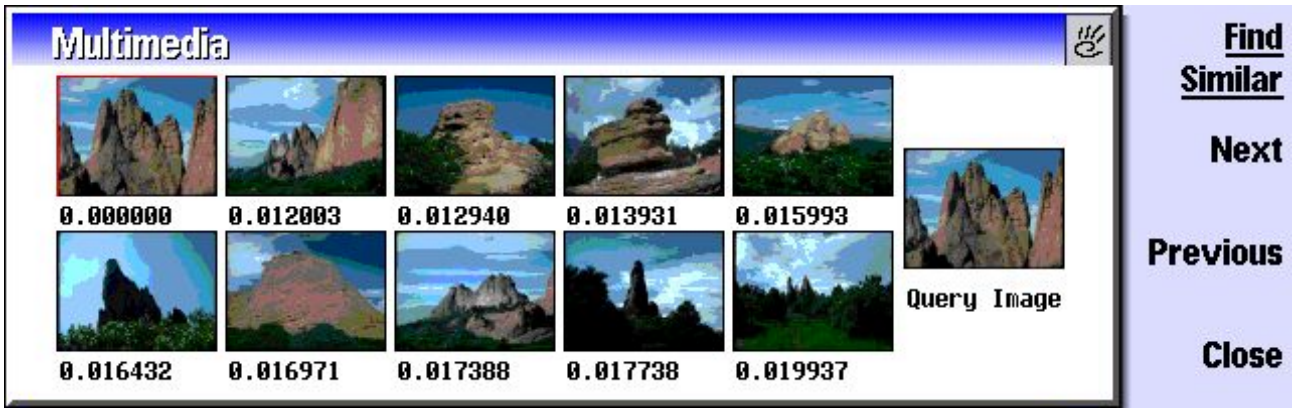

Figure 3: Results of a color histogram-based image query

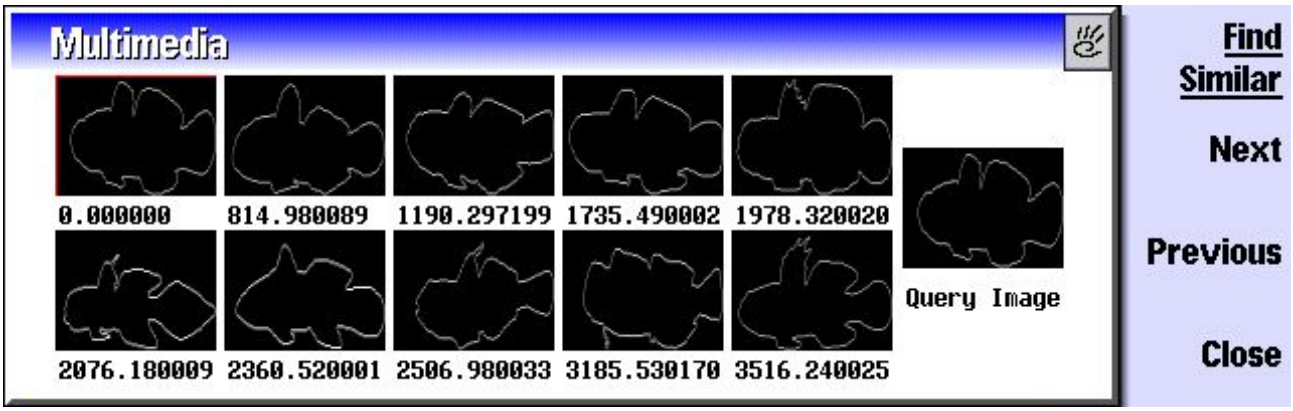

Figure 4: Results of a shape-based image query

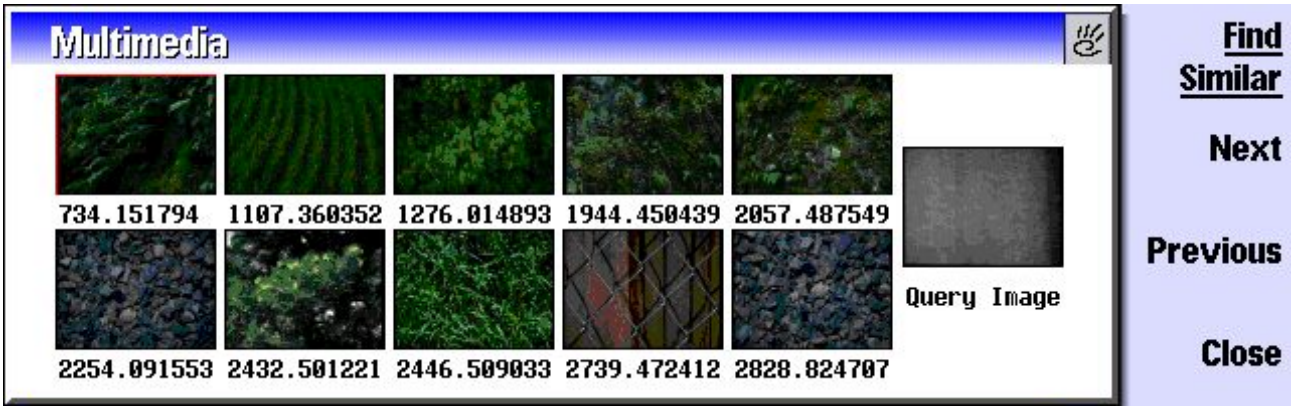

Figure 5: Results of a texture-based image query