University of Southern Queensland

Faculty of Engineering & Surveying

Multimedia Database Search Techniques

A dissertation submitted by

KHAN, Nizam

in fulfillment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Computer Systems)

Submitted: October, 2005

Abstract

From the past decade, the database area has been quite energetic. Several efficient methods are being discovered for managing alphanumeric data. However, when new types of data in the form of image, video and audio are first brought into a database environment, these methods failed to efficiently manage the new kinds of data. Therefore, new multimedia indexing and retrieval methods are required.

This project seeks to investigate possible approaches for deriving multimedia search techniques. The project scope can be categorized into four areas, understanding of different media types, extensive investigation of multimedia database management system, broad research into multimedia document retrieval and implementing some available methods for audio and image retrieval using MATLAB.

In this project, LPC analysis and Filter-Bank analysis are implemented for speaker recognition. In addition, Histogram based image retrieval technique is implemented for color image retrieval. Results are presented for these methods and their performance is discussed.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 & ENG4112 Research Project

Limitation of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports as educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project' is to contribute to the overall education within the student's chosen degree program. This document, the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof G Baker

Dean

Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course of institutions, except where specifically stated.

Nizam Khan

0031231552

_____

Signature

_____

Date

Acknowledgments

This dissertation would never be possible without the generous help and support from many people. I would like to express my deepest respect to my supervisor, Dr John Leis, and thank my family and friends for their encouragement throughout the year.

Nizam Khan

University of Southern Queensland
October 2005

Table of Contents

Abstract ____ *i*

Certification ____ *iii*

Acknowledgments ____ *iv*

List of Figures ____ *ix*

List of Tables ____ *xi*

Chapter 1 ____ *1*

1.1 Introduction ____ **1**

1.2 Project Aim ____ **2**

1.3 Multimedia Database ____ **2**

1.4 Why Multimedia Database? ____ **3**

1.5 Why Multimedia Database Search techniques? ____ **3**

1.6 Overview of the Dissertation ____ **4**

Chapter 2 Multimedia Database Architecture ____ *6*

2.1 Chapter Overview ____ **6**

2.2 Media Types and Multimedia ____ **6**

2.3 Multimedia Database design issues ____ **8**

2.3.1 Data Modeling ____ 8

2.3.2 User interface design ____ 10

2.3.3 Query Support ____ 12

2.3.4 Feature Extraction ____ 14

2.3.5 Indexing and Similarity measurement ____ 15

2.3.6 Multimedia data compression ____ 15

2.3.7 Quality of Services Guarantees ____ 15

2.4 General Model 16

2.5 Chapter Summary 17

Chapter 3 Multimedia Database Search Techniques *18*

3.1 Chapter Overview 18

3.2 Standard Model 19

3.3 Expected Capabilities of Search Techniques 21

3.4 Application Specific Techniques 23

3.5 Search Techniques for Audio Files 24

3.5.1 Speech File Retrieval 24

3.5.2 Music File Retrieval 25

3.6 Search Techniques for Image Files 26

3.7 Search Techniques for Video Files 27

3.8 Chapter Summary 28

Chapter 4 Audio Indexing and Retrieval *29*

4.1 Chapter Overview 29

4.2 Audio Properties and Features 29

4.2.1 Time domain characteristics 30

4.2.1 Frequency domain characteristics 31

4.3 Audio Classification 32

4.4 Speech recognition and Retrieval 34

4.4.1 Filter-bank analysis 35

4.4.2 LPC analysis 36

4.4.3 Vector quantization 37

4.5 Music Recognition and Retrieval 38

4.6 Chapter Summary 39

Chapter 5 Image Indexing and Retrieval *40*

5.1 Introduction 40

5.2 Possible Approaches to Image Retrieval 41

5.2.1 Text-Based Image Retrieval 41
5.2.2 Color-Based Image Retrieval 42
5.2.3 Shape-Based Image Retrieval 43
5.2.4 Texture-Based Image Retrieval 44

5.3 Histogram-Based Color Image Retrieval 45
5.3.1 Color Model 46
5.3.2 Histogram-Based Search 47
5.3.3 Overall Scheme 49

5.4 Chapter Summary 50

Chapter 6 Video Indexing and Retrieval 51

6.1 Introduction 51

6.2 Possible Approaches to Video Retrieval 52
6.2.1 Shot-Based Video Retrieval 52
6.2.2 Motion Information-Based Video Retrieval 55
6.2.3 Object-Based Video Retrieval 55

6.3 Chapter Summary 56

Chapter 7 Performance of Selected Search Techniques 57

7.1 Introduction 57

7.2 Results for Audio Retrieval 58

7.2 Results for Image Retrieval 62

7.4 Computational Efficiency 69

7.5 Discussion of Results 70
7.5.1 Audio Retrieval 70
7.5.2 Image Retrieval 70

7.6 Chapter Summary 72

Chapter 8 Conclusion and Further Work 73

8.1 Accomplishment of Aims 73

8.2 Further Work 75

References 76

Appendix A Project Specification ***79***

Appendix B Source Code ***82***

List of Figures

2.1 Example of creating image data models 10
2.2 User interface of C-BIRD system 11
2.3 Results of Text Search on Artist's Name "Hollowell" 12
2.4 Results of Similarity Search Based on a Hollowell Painting 13
2.5 Basic idea of feature based similarity search 14
2.6 General model of Multimedia Database Architecture 16
3.1 A general multimedia retrieval model 19
3.2 Consequences of a similarity match 20
3.3 Example of annotation based query interface 21
3.4 Example of color layout based image query 22
4.1 Time domain representation of an audio signal 30
4.2 Frequency domain presentation of an audio signal 31
4.3 A possible audio classification process 33
4.4 Pattern recognition model for audio 34
4.5 Bank of filters analysis model 35
4.6 LPC analysis model 36
4.7 Code word in 2 dimensional space 37
4.8 Example of music retrieval system 39
5.1 HSV co-ordinate system and color model 46
5.2 Flow of histogram-based search technique 49
6.1 Interface for searching video shots 53
6.2 shot based retrieval 54
6.3 Browse shot control 54
7.1 The system interface 58
7.2 Audio search options 59

7.3 Inserting an example file to search the audio database 59
7.4 Result of audio search 59
7.5 The system interface 62
7.6 Inserting example file to retrieve similar file 63
7.7 The first input image (616.jpg) 63
7.8 Best 5 retrieval using Histogram Intersection for 616.jpg 63
7.9 Best 5 retrieval using Histogram Euclidian method for 616.jpg 64
7.10 The second input image (620.jpg) 64
7.11 Best 5 retrieval using Histogram Intersection for 620.jpg 64
7.12 Best 5 retrieval using Histogram Euclidian method for 620.jpg 65
7.13 The third input image (810.jpg) 65
7.14 Best 5 retrieval using Histogram Intersection for 810.jpg 65
7.15 Best 5 retrieval using Histogram Euclidian method for 810.jpg 66
7.16 The fourth input image (811.jpg) 66
7.17 Best 5 retrieval using Histogram Intersection for 811.jpg 66
7.18 Best 5 retrieval using Histogram Euclidian method for 811.jpg 67
7.19 The fifth input image (378.jpg) 67
7.20 Best 5 retrieval using Histogram Intersection for 378.jpg 67
7.21 Best 5 retrieval using Histogram Euclidian method for 378.jpg 68
7.22 The sixth input image (379.jpg) 68
7.23 Best 5 retrieval using Histogram Intersection for 379.jpg 68
7.24 Best 5 retrieval using Histogram Euclidian method for 379.jpg 69
7.25 Sets for explaining retrieval effectiveness 70

List of Tables

4.1 Main characteristics of speech and music 32
7.1 Results of audio retrieval using LPC analysis 60
7.2 Results of audio retrieval using Filter-Bank analysis 61
7.3 Retrieval effectiveness 71

Chapter 1

1.1 Introduction

During the last decade, the main focus of multimedia research and development has been on multimedia communication and authoring tools. Since more and more digital multimedia data in the form of images, video and audio is being captured and stored everyday, the focus has shifted over the last few years to efficient and effective storage and retrieval of multimedia documents.

About thirty years ago, text based database started to play an important role in our day-to-day lives. This led to the development of database management systems, which are now one of the most popular computer applications and are used in almost every organization. Due to the differences between the characteristics of text based documents and multimedia documents; a traditional database management system is not capable of handling multimedia data. Therefore, the need for efficient multimedia document management system is becoming a popular research area.

To develop a multimedia database management system, an efficient and effective retrieval of multimedia document system is required. Multimedia documents have special characteristics and requirements that are significantly different from text base

documents. Hence, traditional database search techniques are not suitable for retrieval of multimedia documents.

Consequently, the topic of this research project becomes relevant. This research project is aimed at investigation of current techniques for searching multimedia items from a database. In this project, an attempt is made to implement some of these techniques for an available multimedia database to test their performance.

1.2 Project Aim

This project aims to investigate possible approaches to search documents from a multimedia database and to prototype some systems to test their performance. This project involves the understanding of major signal processing theories, such as pattern recognition theories, media dependent analysis theories and multimedia database management theories for retrieving multimedia documents.

An experimental software framework for audio and image retrieval in MATLAB will be developed. In addition, the efficiency of the techniques will be investigated using speech and image databases.

1.3 Multimedia Database

Multimedia refers to a collection of media types used together. It is implied that at least one type is not alphanumeric data. Therefore, multimedia data, document or item can be described as computer readable representation of multiple media types.

Now, a multimedia database can be defined as a database that contains multiple types of objects such as graphical images, video clips, sound files and the like. This may be, for example, a large collection of movies or music files.

1.4 Why Multimedia Database?

Currently, the society is faced with an explosion of multimedia information. For example, a large amount of images and videos are being created and stored on the internet. Many paintings and pictures in printed from are being converted to digital form for easy processing, distribution and preservation. A large number of medical images are being captured everyday and satellites are producing many more. This trend is going to continue with the advancement of storage and digital technologies.

Alternatively, information is needed in our day to day life in many matters such as for work, entertainment, study and to live. Multimedia data is rich in information. A famous saying can support this statement and it is “A picture is worth a thousand words”. For example, a student may wish to scan in an animal picture and then to be able to retrieve all facts about this type of animal from an educational database. Multimedia database can provide the facts that might include images about different states of those animals as they grow older, video clips about their particular lifestyle and charts about their breeding process. On the other hand, a traditional database can only provide the student with text documents that describes about those animals. Hence, it can be concluded that multimedia database can provide us more information about anything than the traditional database. As a result, multimedia database is needed to cope up with recent explosion of multimedia information.

1.5 Why Multimedia Database Search techniques?

As it is known by now, traditional database search techniques are not suitable for handling multimedia data effectively because of the differences between the characteristics of alphanumeric data and multimedia data. There are a number of differences that can separate multimedia data from alphanumeric data. Some of the key differences are:

- Multimedia data is very data intensive. For example, a 5 minute video file can occupy about 500MB of storage. On the hand, a really big text document can occupy up to 5 MB of storage.

- Audio and video have temporal dimensions and they must be played out at a fixed rate to achieve the desired effect. Alternatively, text documents do not have temporal dimensions to read them.

- The meaning of multimedia data is sometimes fuzzy and subjective for recognition. Instead, text based data have proper definition for recognition.

Therefore, to use multimedia information effectively, multimedia documents should be organized for rapid retrieval on demand. For this organization, effective search techniques are one of the requirements.

1.6 Overview of the Dissertation

This dissertation is organized as follows:
Chapter 2 gives a description of different media types, current multimedia design issues and general multimedia database architecture.

Chapter 3 gives an overview of multimedia database search techniques. This chapter also describes the standard model for multimedia item searching, a number of search techniques for audio, image and video files and expected capabilities of these search techniques.

Chapter 4 investigates about audio type media files. In this chapter, different characteristic of audio files and some important audio retrieval issues, such as speech, speaker and music retrieval are mentioned. Specific techniques for audio retrieval such as the LPC method, the Filter Bank Analysis method and methods for music recognition based on spectral properties are the important part of this chapter.

Chapter 5 explores about image media files. This chapter describes some possible approaches to retrieve image files from a database. Histogram-based image retrieval is briefly discussed.

Chapter 6 discusses the possible approaches to retrieve video files from a database.

Chapter 7 examines the performance of selected search techniques. The computational efficiency is discussed, and results of the experiments are presented.

Chapter 8 concludes the dissertation with the discussion of achievements and suggestions for further work.

Chapter 2

Multimedia Database Architecture

2.1 Chapter Overview

To derive effective multimedia database search techniques, an understanding of multimedia database items and the internal architecture of multimedia database management system is necessary. Initially in this chapter, a description of multimedia data items will be given and later on common multimedia database design issues and architecture will be discussed.

2.2 Media Types and Multimedia

Media refers to the types of information or types of information representation, such as images, audio and video. Thus, multimedia can be defined as a collection of multiple types of media. People who use the term multimedia often seem to have quite different, even opposing, viewpoints. A PC vendor thinks of multimedia as a PC that has sound

capability, a DVD-ROM drive and perhaps the superiority of multimedia-enabled microprocessors that understand additional multimedia instruction. A consumer entertainment vendor may think of multimedia as an interactive cable TV with hundreds of digital channels. But according to Li and Drew (2004, p. 98), "a computer professional should have more application oriented view of what multimedia consist of: applications that use multiple modalities to their advantage, including text, images, drawings, animation, videos, sound and most likely, interactivity of some kind".

Media data items can be classified according to their temporal property: static and dynamic. Static media do not have a time dimension and their contents and meaning does not depend on the presentation time. For example, graphics and still images dose not have any temporal dimension. Dynamic media have time dimensions and their meaning and correctness depend on the rate at which they are presented. For example: audio and video must be played out in a correct rate to maintain its quality.

Images can be grayscale or in color. Different parameters of an image are: image size, the pixel number per line, pixel depth etc. Images usually require a large amount of data. Thus, image compression becomes essential in multimedia applications. Image compression is achieved by exploiting redundancies in images and human perception properties. Human beings can tolerate some information error or loss without affecting the effectiveness of communication. This means that the compressed data does not need to present the original information samples exactly. Some common compression techniques are: spatial sub-sampling, predictive coding, transform coding, vector quantization, fractal image coding, JPEG compression standard, JPEG-2000 etc.

Audio is caused by a disturbance in air pressure that reaches the human eardrum. When the frequency of the air disturbance is in the range of 20 to 20,000 Hz, the human hears sound. One of the parameters is: sound amplitude (threshold of audibility and threshold of pain). Features derived in time domain include: average energy, zero crossing rate and silence ratio. Features derived in frequency domain include: bandwidth, energy

distribution, harmonicity, pitch, and spectogram. The common compression techniques are: quantization, Predictive coding, and MPEG audio (compression technique uses masking property).

A video sequence is displayed as a series of image frames. Each frame is a snapshot of a moment in time of the motion-video data, and is very similar to a still image. When the frames are played back in sequence on a display device, a rendering of the original video data is created. In real-time video the playback rate is 30 frames per second. This is the minimum rate necessary for the human eye to successfully blend each video frame together into a continuous, smoothly moving image. Two major characteristics of video are that it has a time dimension and it takes huge amount of data to represent. In this regard, related issues are: frame rate, bandwidth requirement, interlace, etc. Different video compression techniques are: MPEG-1, MPEG-2, MPEG-4 and MPEG-7.

2.3 Multimedia Database design issues

To devise search techniques for multimedia databases, many researchers recommended several concepts for multimedia database management system. Some of these concepts are described below:

2.3.1 Data Modeling

Grosky (1997) suggested that a data model is a collection of abstract concepts that can be used to represent real world objects, their properties, their relationship to each other, and the operations defined over them. These abstract concepts are capable of representing the unique features of database items. Through the use of these abstract concepts, the unique features of multimedia items can be identified and it can be used for indexing and retrieval.

On the other hand, According to Lu (1999, p. 55) “The role of data model is to provide a framework to express the properties of the data items that are to be stored and retrieved using the system”. The function of this framework should be to allow designers and users to define, insert, delete, modify and search database items and properties. Basically, the additional role of specifying and computing different level of abstraction from multimedia data is done by data model.

The main function of data model is to capture the static and dynamic properties of the database items and to provide a formal foundation for developing the suitable tools needed in using the multimedia data. Object attributes and the relationship between objects could be used as examples of static property. Examples of dynamic properties include operations on objects and user interaction.

Lu (1999) suggested that the data model should meet the following requirements:

1) The data model should be extensible, so that new data types can be added.

2) The data model should be able to represent basic media types and composite objects with complicated spatial and temporal relationships.

3) The data model should be flexible so that items can be specified, queried and searched at different level of abstraction.

4) The data model should allow efficient storage and search.

Fig 2.1: Example of creating image data models

(Source: Grosky 1997, p. 74)

2.3.2 User interface design

User interface is the communication medium between the user and the multimedia database. Hence, user interface establishes the usability of multimedia database. The

main functions of the user interface are to allow the user to insert items into database, to enter queries and to present to the user responses to queries.

According to Lu (1999), a good user interface should meet the following requirements:

i. Provide tools for users to insert database items easily.

ii. Provide tools for users to effectively and efficiently enter queries.

iii. Present query results to the user effectively and efficiently.

iv. Be user-friendly.

An example of C-BIRD user interface is shown below to illustrate the user interface design:

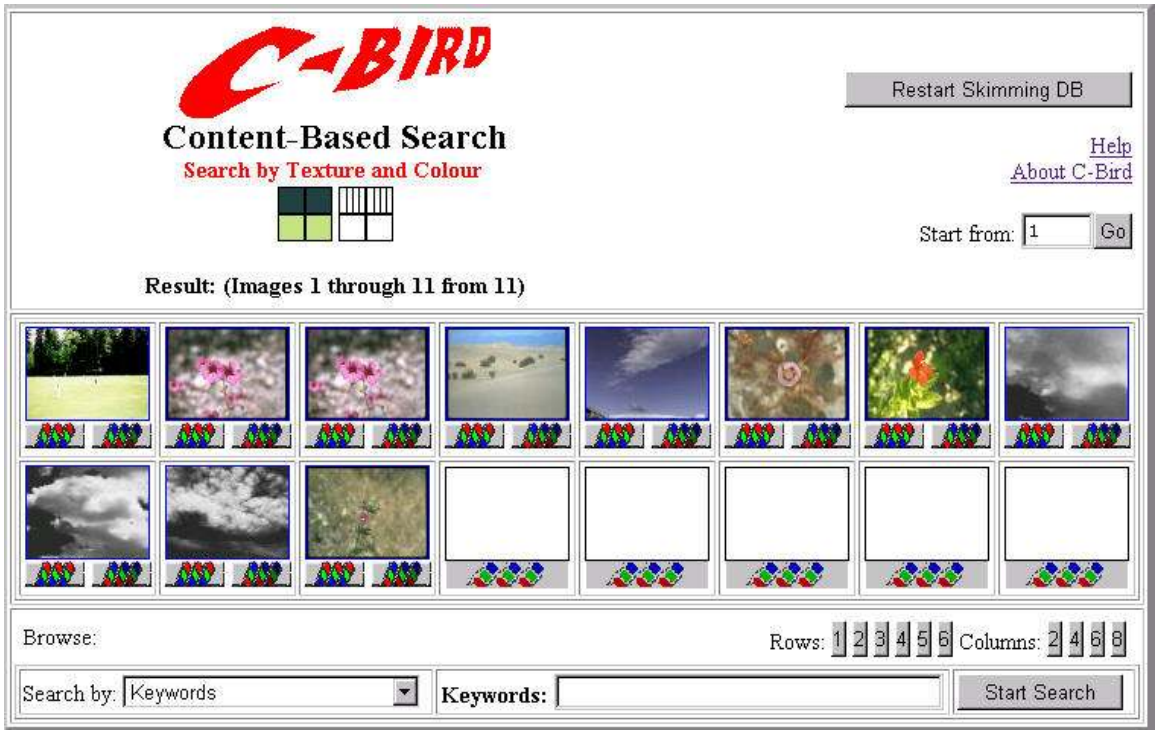

Fig 2.2: User interface of C-BIRD system

(Source: <http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/images/cbird1.jpg>)

2.3.3 Query Support

Fuzziness and diverseness are the properties of multimedia queries. Since user can specify multimedia queries in many different ways, multimedia database management system should provide some form of query support.

Lu (1999) recommended three ways for query support. They are:

a) Searching: Search is one of the essential tasks of all database management systems. In context of multimedia database, search by specification and search by example should be provided. An example of QBIC search engine is shown below to illustrate searching:

Figure 2.3: Results of Text Search on Artist's Name "Hollowell"
(Source: <http://www.cse.unsw.edu.au/~jas/talks/curveix1/pic/qbic1.jpg>)

b) Browsing: Users do not always know what they are looking for but can recognize what they want when they see it. This situation can be resolved by browsing. Browsing should also support search by example capability.

At first, user can be presented with a set of examples. Then the user can start with a very vague query and then can navigate through items based on the results. An example of QBIC search engine is shown in below to illustrate browsing:

Fig 2.4: Results of Similarity Search Based on a Hollowell Painting

(Source: <http://www.cse.unsw.edu.au/~jas/talks/curveix1/pic/qbic2.jpg>)

c) Query refinement: Most results from multimedia queries are fuzzy or not exact. The user interface should provide tools for users to refine their queries based on the results of initial query. Query refinement is normally done based on user's feedback on the relevance of the initial result.

In practice, locating multimedia document should be a combination of search, browsing and query refinement.

2.3.4 Feature Extraction

Feature extraction is the procedure for getting properties of multimedia data items. These properties are going to be the basis for search techniques. Generally there are four levels of features**:** metadata, text annotation, low-level content features, and high-level content features.
According to Lu (1999), feature extraction should meet the following requirements:

- Feature and attributes extracted should be as complete as possible to represent the contents of the information items.
- The features should be represented and stored compactly. Complicated and large features will be time consuming.
- The computation of distance between features should be efficient; otherwise the system response time would be too long.

An example of feature-based search is shown in the figure below:

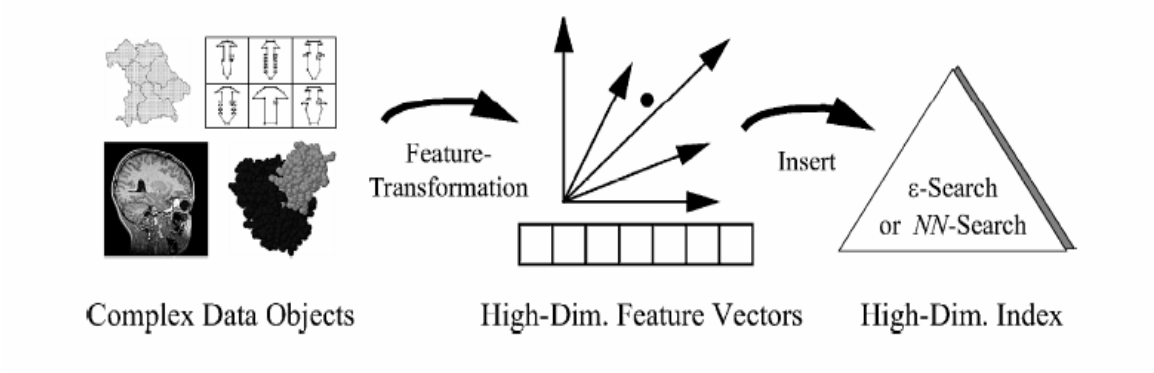

Fig 2.5: Basic idea of feature based similarity search
(Source: Bohm, Berchtold & Keim 2001, p. 324)

2.3.5 Indexing and Similarity measurement

Indexing in multimedia databases should be hierarchical and take place at multiple levels. The levels can be:

1. Application classification
2. Different levels of features
3. Spatial and temporal relationships between objects.

Similarity measurements should be done by considering human judgment. Types of features to describe the objects play an important role for similarity measurement.

2.3.6 Multimedia data compression

The main aim of multimedia data compression has been to compress data as much as possible without considering retrieval and presentation aspects. Most of the multimedia data is stored and captured in compressed form. In order to extract features from these compressed files, they must be decompressed first. This approach is computationally inefficient as compression and decompression may need to be done several times in the feature extraction process. Therefore, there is a need for compression techniques that allow feature extraction directly from the compressed data.

2.3.7 Quality of Services Guarantees

To provide a consistent framework to specify and guarantee the diverse requirement of multimedia management system, QoS has been introduced. QoS is a set of requirement parameters. These requirement parameters are normally specified in two grades: the preferable quality and the acceptable quality. QoS actually defines whether the multimedia database management system maintains a certain standard or not.

2.4 General Model

Multimedia database architecture should be flexible and extensible to support various applications, inquiry types and contents. To satisfy these requirements, common multimedia database should have a number of functional modules. New modules can be added to expand the functionality and existing modules can be deleted or replaced by new ones to improve the functionality.

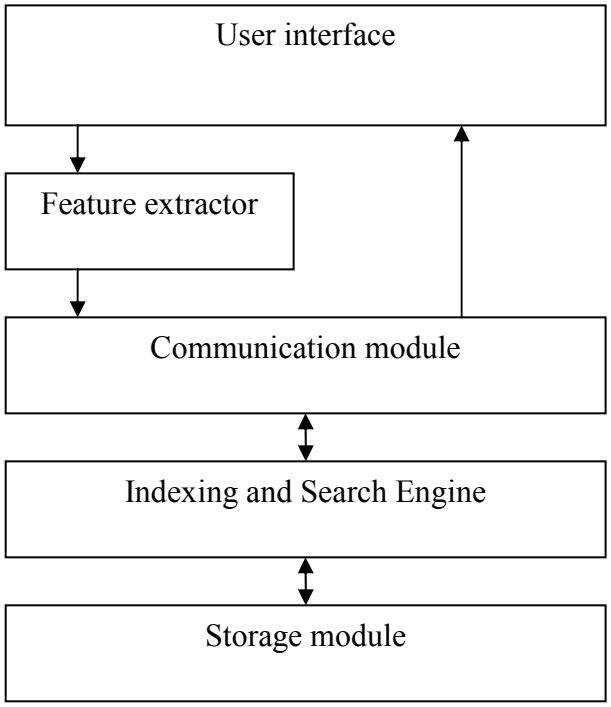

Fig 2.6: General model of Multimedia Database Architecture
(Source: Lu 1999, p. 54)

Figure 2.6 shows the basic architecture of a multimedia database. According to Lu (1999, p 54), "the major functional blocks are the user interface, feature extractor, communication module, indexing and search engine and storage module". The main purposes of these modules can be explained through the operation scenario of

multimedia database. Two major operations of a multimedia database are insertion of new items and retrieval. During the insertion process, the user gives a multimedia item through the user interface. Then the feature extractor creates the feature vector to insert the item in multimedia database. This feature vector and the original item are sent to indexing engine through the communication module. After that, indexing engine provides storage module the multimedia item and the storage module stores that item according to the feature vector.

During the retrieval process, the user issues a query via user interface. Then the feature extractor produces the feature vector of that item. This feature vector is then sent to the search engine through communication module. Subsequently, the search engine provides the storage module with the input feature vector to retrieve similar items. These items are passed to the user interface by the communication module. Then the user interface displays the list of items to the user.

2.5 Chapter Summary

In this chapter, the internal architecture of a general multimedia database management system is analyzed. In addition, the properties of different media types and common multimedia database design issues are also investigated. These concepts are valuable for the understanding of multimedia database management system and will provide the framework for deriving multimedia database search techniques.

Chapter 3

Multimedia Database Search Techniques

3.1 Chapter Overview

This chapter provides a general overview of search techniques used in multimedia database for particular media items. In addition, some important issues relevant to devising search techniques are identified and listed below:

- Standard model for multimedia indexing and retrieval
- Expected capabilities of multimedia search techniques
- Application specific techniques
- Common search techniques for multimedia items

In the following sections, each of these will be discussed in greater detail.

3.2 Standard Model

Traditional database retrieval model retrieve items based on structured data using exact matching. But multimedia retrieval refers to retrieval based on actual media contents such as color and shape, instead of text annotation of the media item. Content based retrieval is normally based on similarity instead of an exact match between a query and a set of database items.

Fig 3.1 provides an overview of standard multimedia retrieval model. In this model, multimedia items are preprocessed to extract features and contents. These multimedia items are indexed according to preprocessed features. During multimedia retrieval, the system takes user's query and processes to extract the main features. These features are then used to compare with the features of indexed multimedia items. The items, whose features are most similar to those of the query, are retrieved and presented to the user.

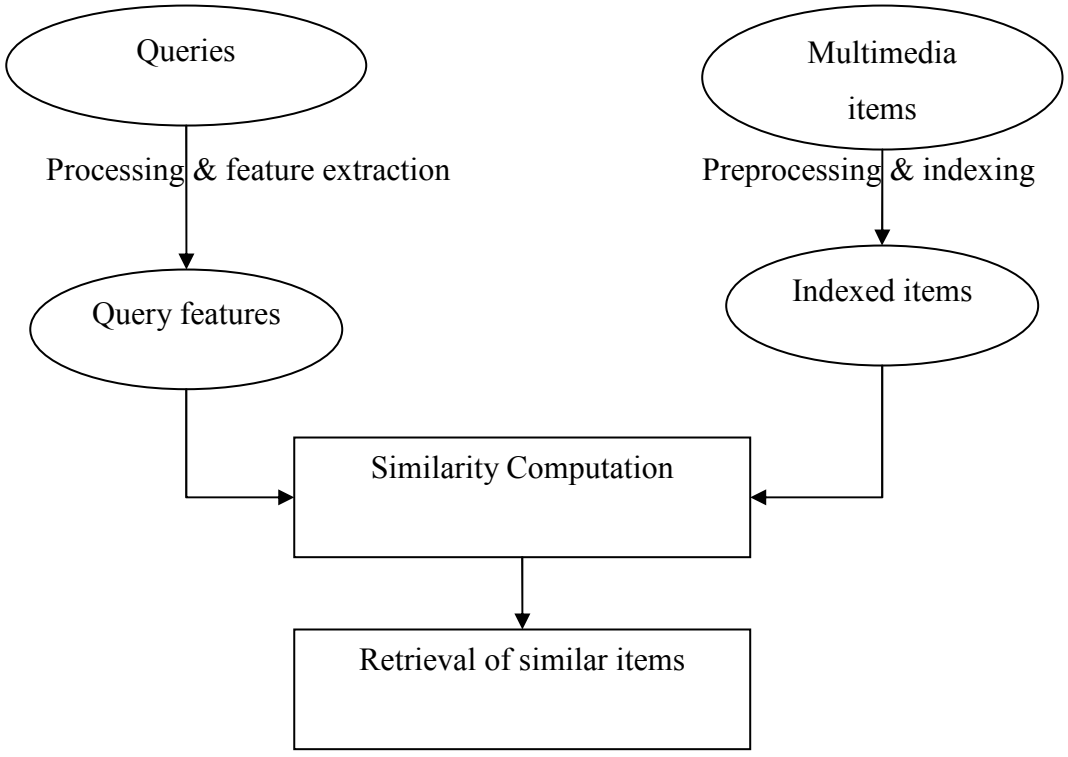

Fig 3.1: A general multimedia retrieval model

Multimedia data is quite different from alphanumeric data. From a presentation viewpoint, multimedia data is huge and involves time dependent characteristics that must be adhered to for coherent viewing. Because of its complex structure, multimedia data requires complex processing to derive semantics from its contents. Real world objects shown in images, videos, animations or graphics and discussed in audio participate in meaningful events whose nature is often the subject of queries. For this reason, multimedia retrieval model uses similarity-based search. The following figure provides an example of the similarity-based search:

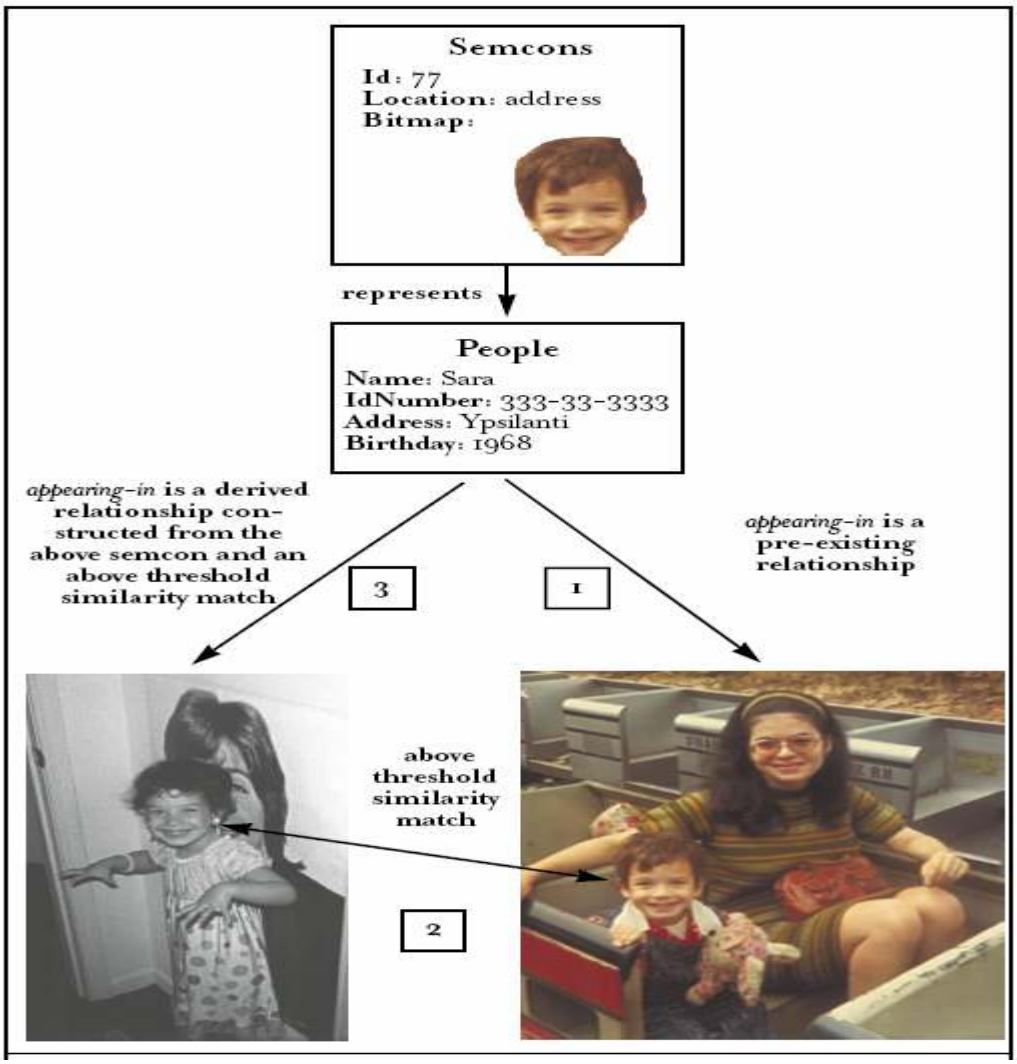

Fig 3.2: Consequences of a similarity match

(Source: Grosky 1997, p. 75)

3.3 Expected Capabilities of Search Techniques

Multimedia database search techniques are expected to be powerful and flexible. Their capabilities can be measured by the types of queries that they can support. Lu (1999) discussed some expected query types and they are as follows.

Metadata-based queries: It refers to the basic properties of database items such as an item name, creation date and modified time. Search techniques should be able to search multimedia items based on the metadata. An example query in multimedia database can be "list images of Tom's vacation". These types of queries can be handled by traditional database search techniques.

Annotation-based queries: It refers to the textual description of the internal content of multimedia items. Queries can be performed by keywords and retrieval is based on the similarity between query and annotation. An example query can be "list all the images showing sunrise". This type of query assumes that items are properly annotated and can be handled by traditional database search techniques. An example of annotation-based queries used in TDT (Topic Detection & Tracking) is shown below:

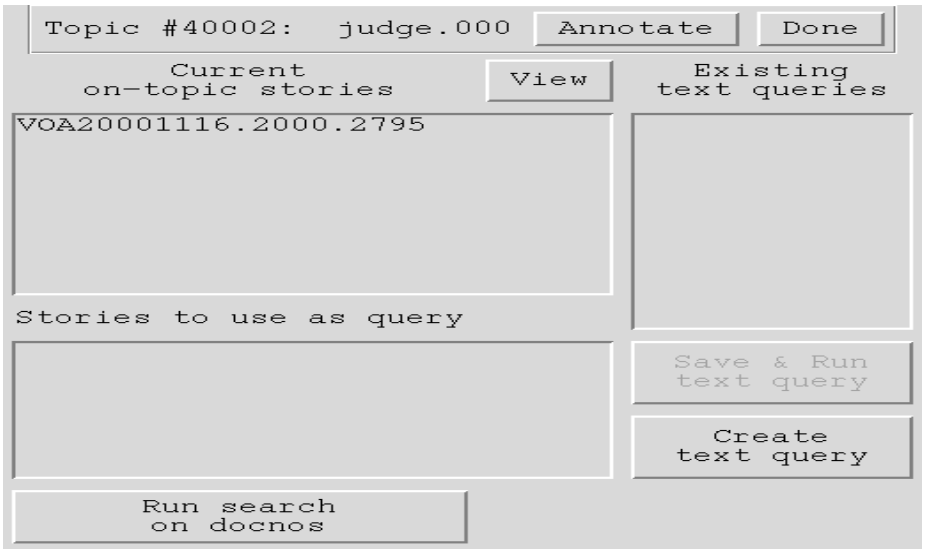

Fig 3.3: Example of annotation based query interface

(Source: <www.ldc.upenn.edu/.../ label_instructions.html>)

Queries based on data patterns or features: It refers to low-level content information about multimedia data, such as loudness, brightness, color percentage and texture description. An example query might be "Search an image with color distribution like THIS". To handle this type of query, low-level content information about multimedia items should be preprocessed, indexed and stored. An example of this type of query used in DB2 Image and Audio Extenders system is shown below:

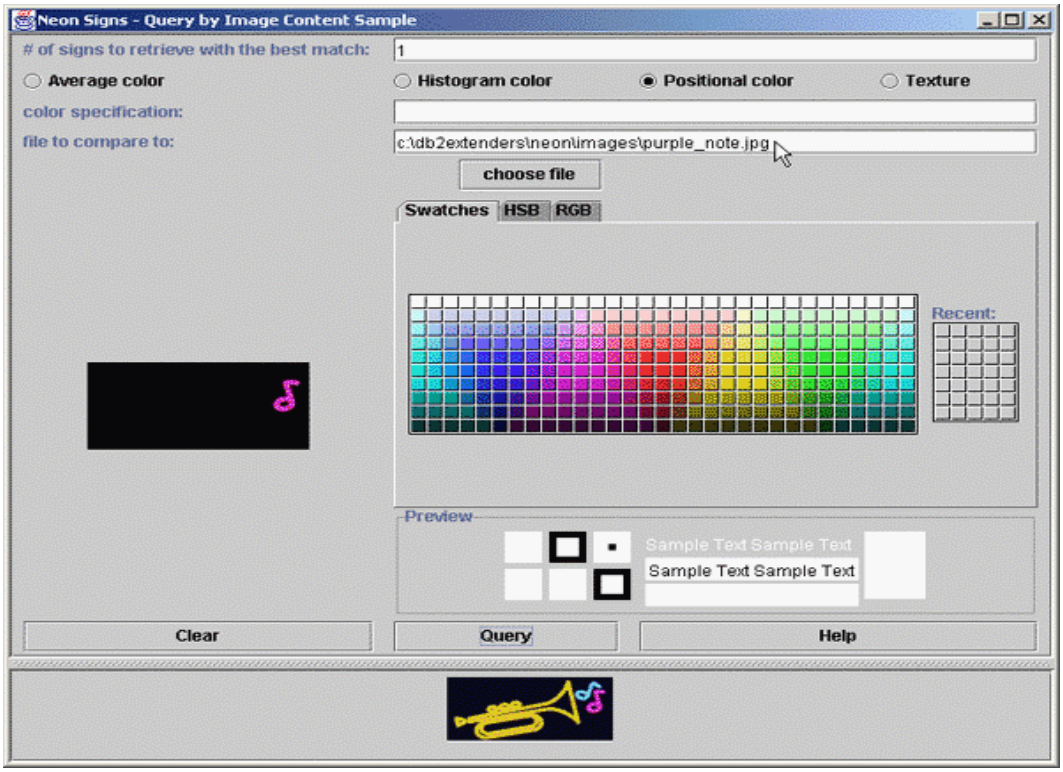

Fig 3.4: Example of color layout based image query

(Source: <http://www-128.ibm.com/developerworks/db2/library/techarticle/0202cox/images/part2fig4.gif>)

Query by example: It refers to the use of multimedia objects such as images, sketches and sound files. User can search for images using a particular sketch. In this regard queries can be complicated by spatial and temporal properties of multimedia items.

In short, these types of queries are expected by a multimedia system to provide the user with maximum chance to find the item of interest.

3.4 Application Specific Techniques

Multimedia database search techniques can be used in many different applications. For example, searching can be based on very specific and detailed information such as the size of an object and the face pattern of a person. Hence, multimedia search techniques are expected to support diverse type of queries and thus will have wide range of applications. Some of these application areas are discussed below:

Multimedia search techniques can be used in medical analysis. For example, a doctor can use the ultrasound image of a patient to retrieve images with a comparable degree of left ventricular hypertrophy from an ultrasound image database. Currently, the School of Electrical and Computer Engineering at Purdue University designed a human-in-the-loop CBIR (Content-Based Image Retrieval) system for medical images.

Alternatively, multimedia search techniques can be very useful in security issues. Law enforcement agencies can use an image of a criminal to retrieve all the necessary information from a security information database. For example, nevenvision (a security company) devised a system that recognizes faces from video images.

Moreover, multimedia search techniques can be very helpful for students to get any information from an educational database. For instance, a student scans an animal picture and wants to retrieve all facts about this type of animal from an animal database.

However, multimedia search techniques can also be useful for journalists. A reporter writing an article on a person can retrieve that person's pictures and associated information that has appeared in newspapers and television from an information database is an example.

Entertainment is another vast area. In this area, multimedia search techniques can be implemented to create a number of applications. For example, Video-on-Demand servers provides user with video items based on user's content-based query.

Multimedia search techniques can be used in trademark registration, copyright and piracy protection. For instance, an officer processing a trademark application wants to determine whether a similar trademark has been previously registered. This problem can be solved by implementing multimedia search techniques in a trademark database to compare the most similar existing trademark to the new trademark.

In summary, it can be inferred that multimedia search techniques can be used in many different areas. Implementing these techniques in different systems can make our life lot easier.

3.5 Search Techniques for Audio Files

According to Lu (1999), the first step to search an audio data file is to identify what kind of audio file it is. He classified audio data in two ways, such as: 1) Speech audio data 2) Music audio data. He recommended an audio data can easily be classified to speech or music data file according to the property of audio data file. Since speech audio file has low bandwidth, higher silence ratio and higher variability in zero crossing rates, it can easily be differentiated from a music data file. On the other hand, music data file has properties like high bandwidth, lower silence ratio, and lower variability in zero crossing rates, regular beats. These differences in properties can be a step in deriving search techniques. Some of the common techniques for retrieving audio files are as follows.

3.5.1 Speech File Retrieval

Broadly speaking, there are three approaches to speech file recognition (Rabiner & Juang, 1993, p. 42), namely:

- The acoustic-phonetic approach
- The pattern recognition approach
- The artificial intelligence approach.

Among these approaches, pattern recognition approach is the popular one and widely implemented. In this approach, spectral analysis is used as parametric representation of speech files. According to Rabiner and Juang (1993), this pattern recognition approach consists of three paradigms, namely:

I. Feature measurement using some type of spectral analysis techniques, such as a filter bank analyzer, a linear predictive coding analysis or DFT analysis.
II. Pattern classification, in which unknown test pattern is compared with each database items and a measurement of similarity between the test pattern and database items are computed.
III. Decision logic, in which the database items similarity scores are used to decide which database item best matches the unknown test pattern.

3.5.2 Music File Retrieval

In order to derive techniques for a particular music file, Lu (1999) discussed three different techniques. They are:

- Music indexing and retrieval of structured music and sound effects.
- Retrieval based on a set of pre-defined features.
- Music retrieval based on pitch (pitch tracking).

Among these techniques, first technique is used for structured music MIDI, which represents music as a number of notes and control commands. Using music notes as a unique feature of music file, it is easy to search a database for a particular music file. In the second technique, a set of acoustic features is extracted for each music file. Then the similarity between query and each of the stored music pieces is calculated based on the closeness between their corresponding feature vectors. Finally, in the third technique, pitch tracking is used to retrieve music files. A pitch tracking is a symbolic

representation of music files. This symbolic representation is then used to find similar music files from a database.

3.6 Search Techniques for Image Files

For searching an image file contained in a database, Li and Drew (2004) suggested several techniques. They used some of those techniques in their demo image search engine, named C-BIRD. C-BIRD uses different statistical approaches to find image data file from an image database. These techniques are:

- Color histogram
- Color density
- Color layout
- Texture layout.

It also included some approaches according to:

  - Illumination invariance
  - Object model

Among these approaches, Color histogram is a common statistical analysis technique for image data files. A color histogram counts pixels with a given pixel value in red, green and blue (RGB). So, the color histogram of each database items should be pre-calculated and then these features of each item can be compared with the query item to compute similarity.

In the color layout technique, Li and Drew (2004) suggested that an image can be retrieved by setting up a scheme of how colors should appears in the image, in terms of coarse blocks of color. User can set up a combination of properties for an item and then search in the database for an item that has similar properties.

In color density technique, Li and Drew (2004) recommended to set up a technique that let user selects the percentage of the image having any particular color or set of colors, using a color picker and sliders. Then database can be searched for a similar item.

On the other hand, Castelli & Bergman (2002) recommended using texture features for image retrieval. He discussed Tamura's features such as coarseness, contrast and directionality can be the basis of image retrieval. These features capture the high-level perceptual attributes of a texture well and are useful for image browsing.

Then again, Castelli & Bergman (2002) suggested image retrieval based on shape. There are several approaches to implement this technique, such as Invariant moments, Fourier descriptor method, Histogram of significant edges and Region based shape similarity.

In summary, it can be concluded that there are a huge number of image retrieval techniques available. Some of these techniques are computationally expensive and hard to implement and some of them are pretty straightforward. Hence, the selection of search technique should be based on the database environment.

3.7 Search Techniques for Video Files

Video is information rich. A complete video may consist of subtitles, soundtrack and images recorded or played out continuously at a fixed rate. Lu (1999) discussed some methods for video indexing and retrieval. These methods are as follows:

Meta-data based method: In this method, video files can be retrieved using traditional database search techniques. Common metadata are the video title, author/producer/director and date of production.

Text-based method: Video file is retrieved based on associated subtitles using the traditional database search techniques. Temporal information is usually included to associate text with related frames.

Audio-based method: Video file is retrieved based on associated sound tracks using speech or music recognition. Again, temporal information is usually included to associate the sound with frames.

Content-based method: There are two general approaches to content based video retrieval. First approach, video is treated as collection of images and then image retrieval method is used to retrieve the video file. The second approach divides video sequence into groups of similar frames and retrieval is based on representative frames for these groups. This approach is called shot-based video retrieval.

Integrated approach: Two or more of the methods mentioned above can be combined to provide more effective video retrieval techniques.

3.8 Chapter Summary

In this chapter, the fundamental model for deriving a multimedia search technique is discussed. More focus has been placed on major issues relevant to deriving search techniques such as

- Expected capabilities of search techniques.
- Requirement of application specific techniques.
- Common search techniques for retrieval of audio, image and video files.

Chapter 4

Audio Indexing and Retrieval

4.1 Chapter Overview

This chapter is devoted to the automatic indexing and retrieval of audio. The fundamental issues relating to audio retrieval are identified and listed below:

- Audio properties in time and frequency domain.
- Speech recognition and retrieval.
- Music recognition and retrieval.

In the following sections, each of these issues will be briefly discussed.

4.2 Audio Properties and Features

As it has already been discussed in Chapter 2, audio is caused by a disturbance in air pressure that reaches the human eardrum. When the frequency of the air disturbance is in the range of 20 to 20,000 Hz, human usually hears sound. Audio signals can be presented in two different ways to analyze the signal characteristics. These characteristics can be used for audio retrieval. The two different ways of audio signal analysis are as follows.

4.2.1 Time domain characteristics

Time domain characteristics representation of an audio signal, is the most basic signal representation technique available. In time domain, audio signal is represented as amplitude varying with time. Figure 4.1 shows a typical digital audio signal in time domain. In this figure, zero represents silence. The signal value varies from positive to negative depending on the sound pressure. From this time domain presentation, a number of different characteristics of an audio file such as average energy, zero crossing rate and silence ratio can be calculated.

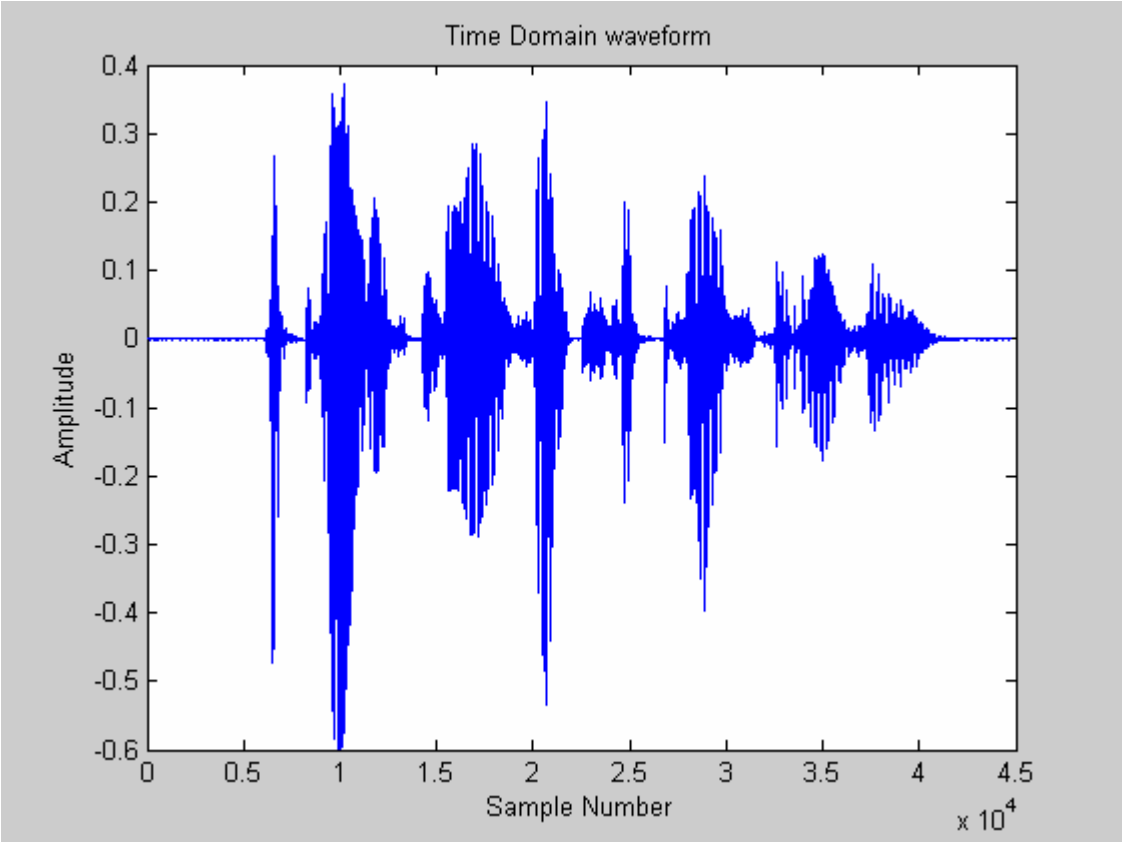

Fig 4.1: Time domain representation of an audio signal

The average energy indicates the loudness of audio signal. Then, zero-crossing rate indicates the frequency of signal amplitude sign change. And finally, silence ratio indicates the proportion of the signal that is silent.

4.2.1 Frequency domain characteristics

In this domain, the signal represents the frequency components and frequency distribution of an audio signal. This presentation is derived from the Fourier transform of time domain signal. An example of frequency domain signal is shown below:

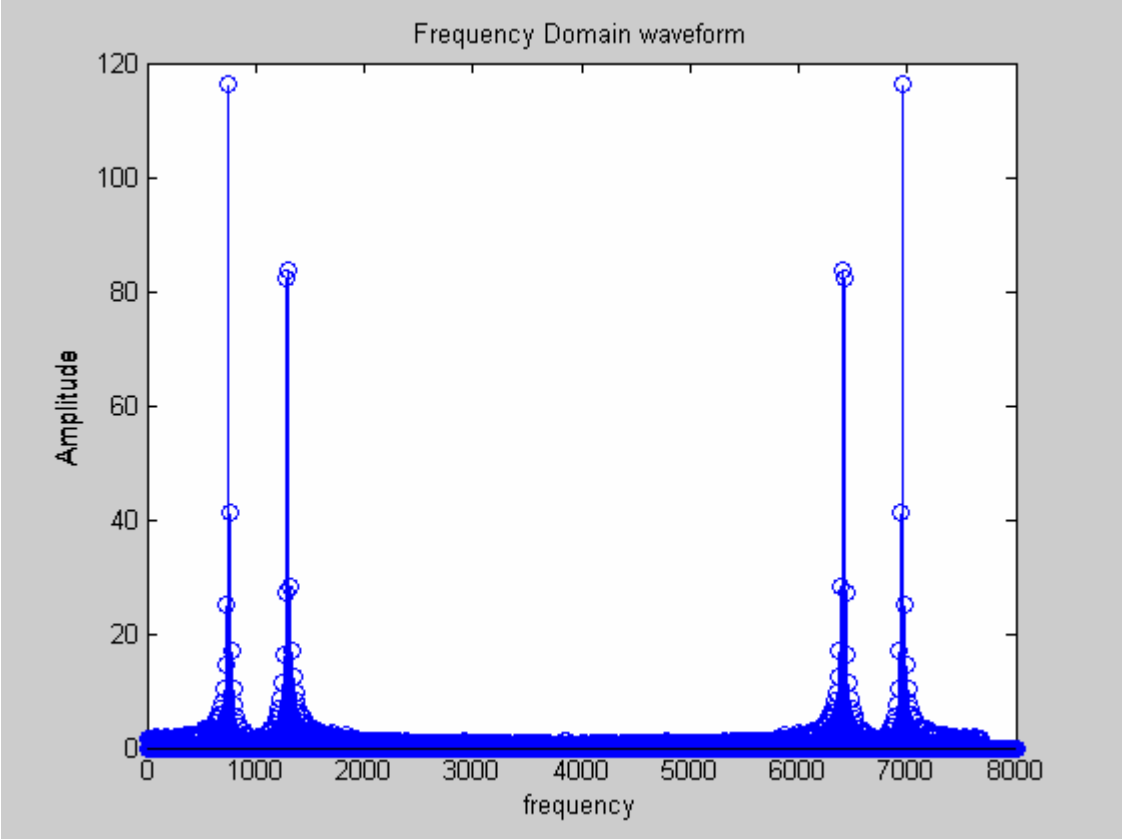

Fig 4.2: Frequency domain presentation of an audio signal

From this presentation, bandwidth, energy distribution (brightness), harmonicity and pitch can be calculated for an audio signal. Bandwidth indicates the frequency range of a sound. Energy distribution provides with spectral energy distribution. While on the other hand, Harmonicity describes the rhythmic pattern of an audio signal. And pitch defines the period sounds, such as those produced by musical instruments and the voice. These properties can then be used to identify a unique type of audio file.

4.3 Audio Classification

Lu (1999) suggested classifying audio signal in two ways. They are:

- Speech audio
- Music audio

Table 4.1 summarizes the major characteristics of speech and music audio. It can be deduced from the table that speech and music signal are different from each other. These differences in properties can be the basis in deriving search techniques.

Features	**Speech**	**Music**
Bandwidth	0 - 7 kHz	0 – 20 kHz
Spectral Centroid	Low	High
Silence Ratio	High	Low
Zero-crossing Rate	More variable	Less variable
Regular Beat	None	Yes

Table 4.1: Main characteristics of speech and music

(Source: Lu 1999, p. 113)

Audio classification can be done in two ways. First method is step-by-step classification in which, each audio feature is used separately to determine if an audio peace is music or speech. In this method, each feature is seen as a filtering or selection criterion. At

each filtering step, an audio peace is determined as one type or another. Figure 4.3 depicts a possible filtering process. The second method is feature vector based audio classification. In this method, the feature vectors of a test audio file are calculated. Then it is compared with trained feature vector of speech and music files. The input is classified into the class from which the input has least vector distance. This approach assumes that audio files of the same class are located close to each other in the feature space and audio files of different classes are located far apart in the feature space.

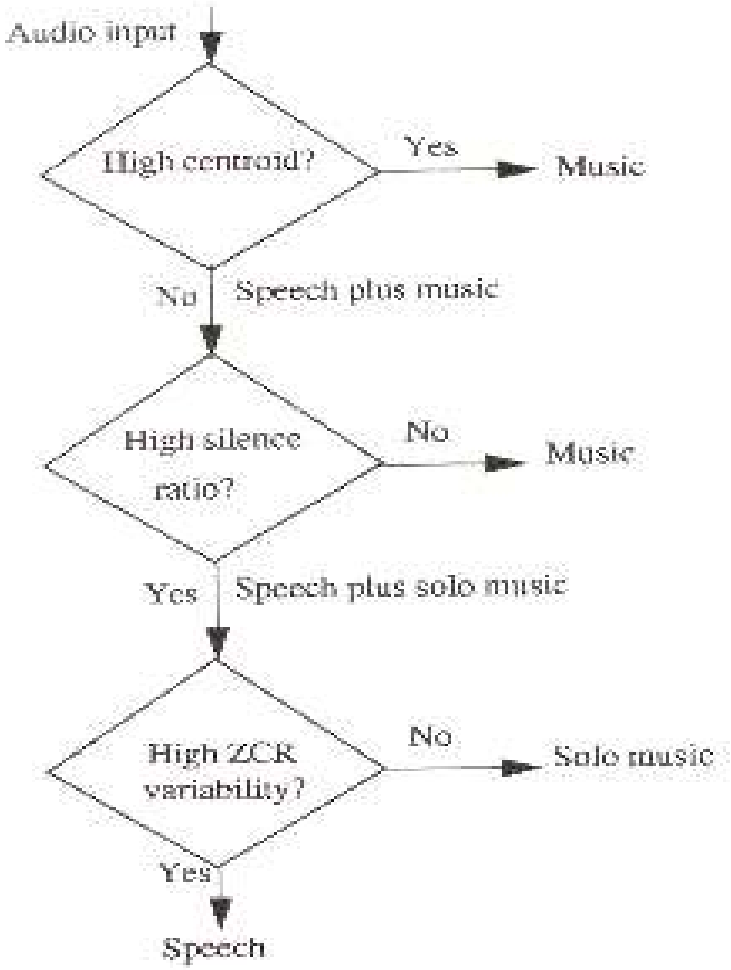

Fig 4.3: A possible audio classification process

(Source: Lu 1999, p.115)

4.4 Speech recognition and Retrieval

In general, the automatic speech file recognition and retrieval is a pattern recognition problem. For this project, pattern recognition model is followed, as an attempt will be made to implement this technique. The brief description of this model is as follows.

The general diagram of a pattern recognition model is shown below:

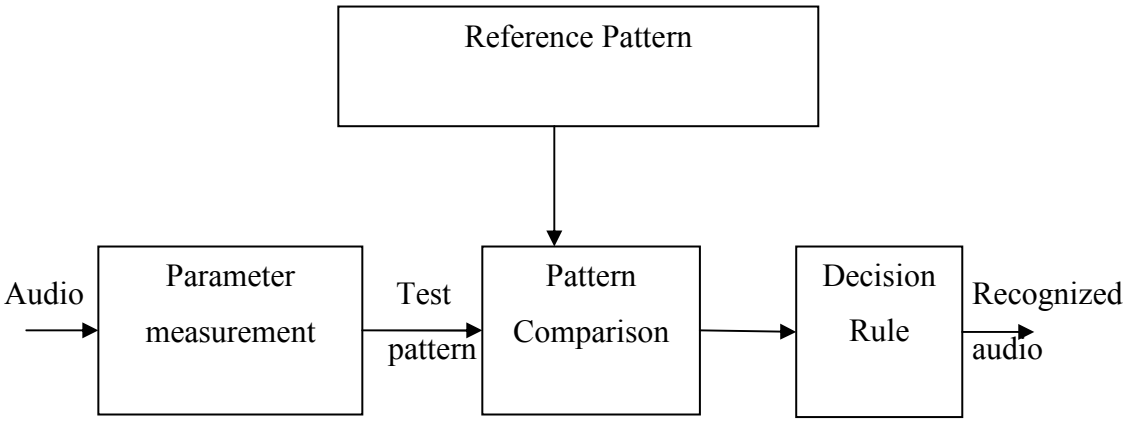

Fig 4.4: Pattern recognition model for audio

In this method, the first job is to verify the type of audio the user is interested in. At first the system requires the user to insert the kind of audio file he is searching for or the system automatically detect the kind of audio data when the user searches by an example file. Then the system extracts the features from the example data or user input query. Rabiner & Juang (1993) suggested these extractions can be done by a bank-of-filter model or LPC model for speech audio file.

4.4.1 Filter-bank analysis

In the bank-of-filter model, the speech signal is passed through a bank of Q band pass filters whose coverage spans the frequency range of interest in the signal (100-3000 Hz for telephone quality signal, 100-8000 Hz for broadband signals). The individual filters can and generally overlap in frequency. These bank-of-filters processes the speech signal to produce a rough approximation of signal spectral shape while smoothing out the harmonic structure if any. These spectral representations can then be used for similarity measurements.

The general model of bank-of-filters is schemed below:

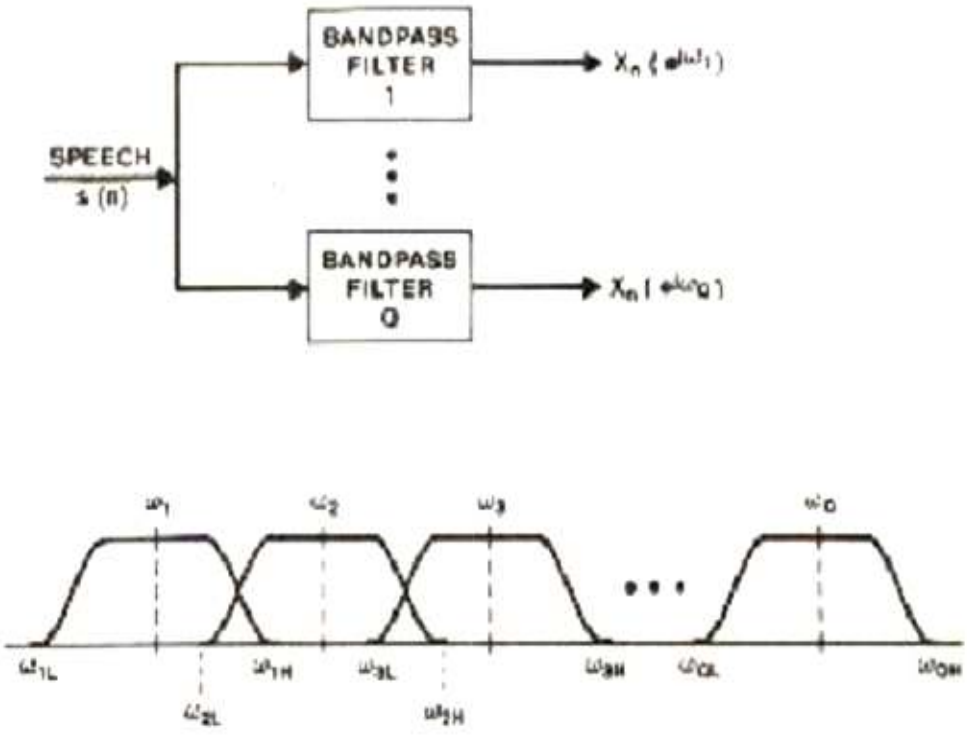

Fig 4.5: Bank of filters analysis model
(Source: Rabiner & Juang 1993, p. 72)

4.4.2 LPC analysis

In LPC analysis, spectral analysis is performed on block of speech (speech frames) with an all-pole modeling constraint. The output of the LPC spectral analysis block is a vector of coefficients (LPC parameters) that specify the spectrum of an all-pole model that best matches the signal spectrum over the period of time in which the frame of speech sample was accumulated (Rabiner & Juang 1993, p. 72). The general model for LPC analysis is shown below:

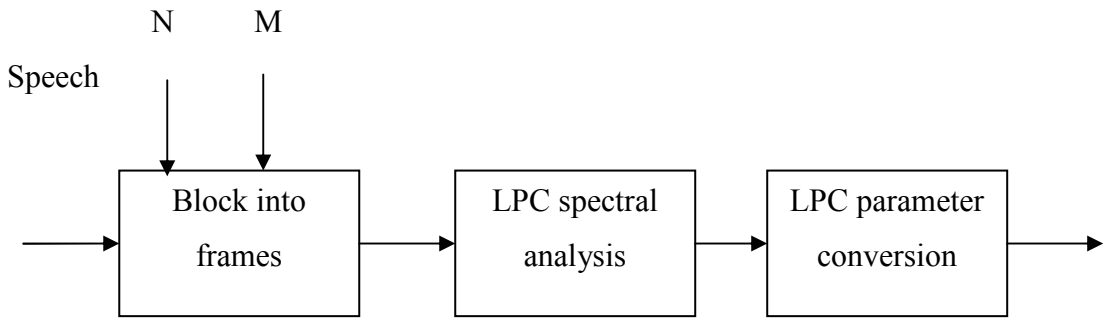

Fig 4.6: LPC analysis model

These two methods have proven themselves to give highest performance in practical speech-recognition system. By using any of these two models speech audio file can be analyzed in low-level. Hence, using these feature extraction procedures, test patterns of input audio and reference patterns for database items are computed.

The results of either filter bank or LPC analysis are a series of vectors characteristic of the time varying spectral characteristics of the speech signal. But these spectral parameters can occupy a lot of memory and comparing these parameters can be computationally expensive. For this reason, vector quantization is used to reduce the raw spectral representation of speech signal. Vector quantization is used in many applications such as image and voice compression and voice recognition (in general statistical pattern recognition).

4.4.3 Vector quantization

Vector quantization process maps k-dimensional vectors in the vector space R^k into a finite set of vectors Y = {y_i: i = 1, 2... N}. Each vector y_i is called as codeword. And the set of all the codewords is called a codebook. Associated with each codeword, y_i, is a nearest neighbor region called Voronoi region, and Qasem (1997) described it by:

$$V_i = \left\{ x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \text{for all } j \neq i \right\}$$

The set of Voronoi regions partition the entire space R^k such that:

$$\bigcup_{i=1}^{N} V_i = R^k$$

$$\bigcap_{i=1}^{N} V_i = \phi \qquad \text{for all } i \neq j$$

Figure 4.7 shows some vectors in space. Associated with each cluster of vectors is a representative codeword. Each codeword resides in its own Voronoi region. These regions are separated with imaginary lines in figure 4.7 for illustration. Given an input vector, the codeword that is chosen to represent it is the one in the same Voronoi region.

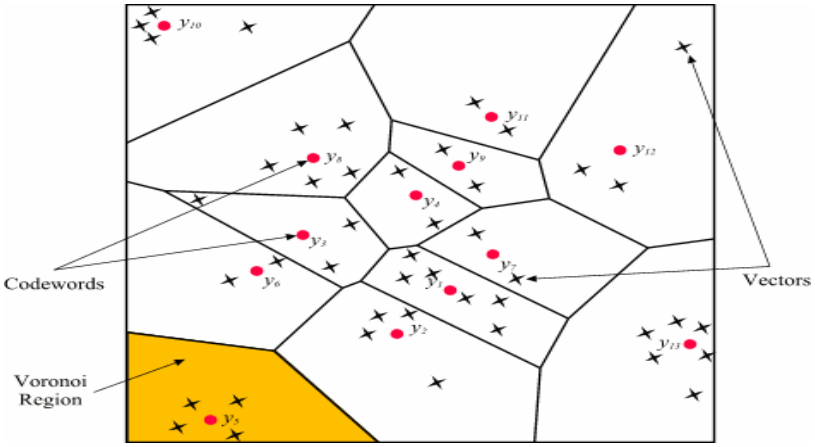

Fig 4.7: Code word in 2 dimensional space. Input vectors are marked with an x.
(Source: Qasem, 1997)

The representative codeword is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2}$$

where x_j is the jth component of the input vector, and y_{ij} is the jth is component of the codeword y_i.

After coding all the reference patterns in the codebook, the system then provides vector quantizer with the spectral vector of input speech signal. This spectral vector is then compared with the code words to measure similarity. The codeword that returns least distance is assumed to be similar speech file and it is retrieved. Thus vector quantizer provides a decision about the similarity computation and a similar speech file is recognized.

4.5 Music Recognition and Retrieval

In general, research and development of effective techniques for music indexing and retrieval is still in its early development stage. Lu (1999) recommended retrieving music files based on a set of features.

In this method, the extraction of features of music signal is necessary. At first, the extraction of features such as: loudness, pitch, brightness, bandwidth, harmonicity are done. These features of sound vary over time and thus are calculated for each frame. Each feature is then represented statistically by three parameters: mean, variance, and autocorrelation. The Euclidian or Manhattan distance between the query vector of each stored music is used as the distance between them. The reference music file that returns least distance assumed to be similar file and it is retrieved. Thus this music retrieval method is implemented. A good example using this approach is the work carried out at

Muscle Fish LLC. They derived a system named SOUNDFISHER that retrieve and sorts according to the content of music files.

SOUNDFISER interface window is shown below to illustrate the retrieval system.

Fig 4.8: Example of music retrieval system

(Source: <http://www.soundfisher.com/html/tour1.html>).

4.6 Chapter Summary

In this chapter, some methods for audio retrieval are discussed. In addition, the overview of audio signal properties and audio classification is illustrated through figures and flowchart respectively. A list of summary of the discussed methods is listed below:

- Filter-Bank analysis.
- LPC analysis.
- Vector quantization.

Chapter 5

Image Indexing and Retrieval

5.1 Introduction

Digital images have predominant position among multimedia data types. Unlike video and audio, which are mostly used by the entertainment and news industry, images are central to a wide variety of fields ranging from art history to medicine, oil exploration and weather forecasting (ed. Castelli and Bergman 2002, p. 1). Image retrieval plays a variable role in numerous human activities, such as law enforcement, earth science, urban planning as well as sports, news casting and entertainment.

This chapter provides an overview of image indexing and retrieval. Different approaches to image retrieval are discussed and histogram-based color image retrieval is briefly described, as an attempt will be made to implement this technique during this project.

5.2 Possible Approaches to Image Retrieval

More research has been carried out in image indexing and retrieval than in audio and video indexing and retrieval. A number of practical techniques and commercial products in image indexing and retrieval are now available. Among them there are four main approaches, which are based on structured properties, object recognition, text and low-level image features (Lu 1999, p.131). Their fundamental description is as follows:

5.2.1 Text-Based Image Retrieval

In this approach, free text is used for image description. User inserts the query in the form of keywords with or without the help of Boolean operators. This retrieval technique uses the traditional database search techniques to retrieve images. It calculates the similarity by comparing the query text and the text description of database images.

To implement this method, the text description of all database images has to be inserted manually. This process is called text annotation. The main issue relevant to text annotation is how to describe an image completely. For example, ‘child’, ‘man’ and ‘woman’ are the subclasses of the more general term ‘people’ and ‘human being’ (Lu 1999, p.133). This specification is known as domain knowledge. Using this domain knowledge, text based retrieval method can easily be implemented.

However, the text description of images cannot always be complete. User may want to search for a file only knowing the subjective feature of the image. So in this method, a relevant feedback method should be available to user. Relevance feedback is very effective as users can determine whether the image is relevant very quickly (Lu 1999, p. 133). Since text description is not always complete, relevance feedback can be used to modify the description of images to make it more accurate.

Main advantages of this method are, very easy to implement, computationally fast and efficient, able to capture high-level abstractions in images. Though this method can not capture low level contents of images, it is widely used by many web search engines.

5.2.2 Color-Based Image Retrieval

Recent progress in automatic feature extraction and image content analysis have enabled the development of new functionalities for searching and accessing images based on perceptual features such as color. Although color distribution and color layout often provide poor characterization of images, color based image retrieval appears to be effective in many situation.

There is number of image search engines available which uses color based image retrieval technique. The first seminal work was carried out by IBM in the QBIC project. This project developed a novel method of prefiltering of queries that greatly reduces the number of target images searched in similarity queries (ed. Castelli and Bergman 2002, p. 285). Then much work has been done in this area such as in MIT photobook project, VisualSEEK, WebSEEK and ImageRover. These systems use color-based search techniques to retrieve images and some of them are really efficient.

The basic idea of color-based image retrieval is to retrieve images that have perceptually similar colors to the user's query image or description. Many techniques have been proposed but their ideas are similar.

Main idea for color-based image retrieval: Images in any database are usually represented by three channels of the color space chosen. There are a number of color spaces available such as, RGB, HSV, HIS and YIQ. At first, color space is partitioned to produce the total number of discrete color combinations. Then a color histogram H(M) is calculated which is a vector (h_1, h_2… h_j , …,h_n) where element h_j represents the number of pixels in image M falling into color j. This histogram is then stored as feature vector of a particular image.

During image retrieval, a histogram is calculated based on user's query. Then the distances between the histograms of the query and database images are calculated. The database image that returns least distance assumed to be the similar image and retrieved.

Although this technique is widely used by many search engines, this technique suffers from a number of limitations. The first limitation of this color-based image retrieval technique is that the similarity between different colors is ignored. According to Lu (1999), if all images have N pixels, then the distance between two images is always less than or equal to 2N. Their distance will be maximal if two images have no common colors after color discretization. That means two images with perceptually similar color but with no common color will have maximum distance. So, this technique can ignore similar images for this limitation. Then the second major limitation is that it ignores spatial relationship among pixels. But this limitation can be solved by image segmentation. And the third limitation is that usually the chosen color space is uniformly quantized, although image pixel colors are not uniformly distributed in the color space. To overcome this problem Wan and Kuo (1996) proposed to quantize colors non-uniformly.

5.2.3 Shape-Based Image Retrieval

The shape-based image retrieval is less developed than color or texture based image retrieval, mainly because of the inherent complexity of representing it. Yet, shape-based retrieval has the potential of being the most effective search technique in many application fields.

Although it is difficult to characterize and manipulate, shape is a significant cue for describing objects. Shape-based retrieval can be very useful in areas such as trademarks and logos, medical domain, architectural domain, law-enforcement and earth science applications. There are a number of ways to represent shape and similarity measurement. Some of these techniques are described below:

Lu (1999) discussed a technique named Fourier descriptor method. In this method, a shape is first represented by a feature function called a shape signature. Then discrete Fourier transform is applied to the signature to obtain Fourier Descriptor (FD) of the

shape. These FDs are used to index the shape and for calculation of shape. Then radius based signature is used for shape classification. During image retrieval, user provides a shape or an example image for shape representation and FD is calculated for input shape. Then this query shape is compared using Euclidian distance with the indexed FDs. The image that returns least distance is assumed to contain the similar shape as input.

Jain & Vailaya (1995) proposed to represent the shape information of an image on the basis of its significant edges. In this method, a histogram of edge directions is calculated and is used to represent the shape features. During retrieval, histogram of edge directions is calculated for user's image. Then it is compared with histograms of edge directions of database images. For comparing usually L-1 metric is used. Since matching the histograms of the edge directions is inherently not rotation or scale invariant, a normalization process in needed (Lu 1999, p. 146). This approach is slow and not efficient for all images.

5.2.4 Texture-Based Image Retrieval

Texture is a property of image regions, as is evident from the pictures of water, grass and a bed of flowers. Texture has no universally accepted formal definition; although one can think of a texture as consisting of some basic primitives whose spatial distribution in the image creates the appearance of a texture (ed. Castelli and Bergman 2002, p. 313).

Many researchers proposed many different ways to extract the texture property of images. Among these researchers Julesz's Co-occurrence Matrices method and Tamura's features is in fact popular. A short description of these methods is described below:

Texture manifests itself as variations of the image intensity within a given region. Julesz (1975) suggested that human texture discrimination is based on the second order statistics of image intensities. A popular texture descriptor emerged from Jultesz's (1975) idea named Co-occurrence matrices. A Co-occurrence matrix counts how often pairs of grey levels of pixels, separated by a certain distance and lying along certain direction, occur in an image (ed. Castelli and Bergman 2002, p. 318). This co-occurrence matrix is then used for texture presentation. Thus it can be used for texture based image retrieval. In addition, this method is expensive to compute; hence it is rarely used in image database application.

Tamura, Mori & Yamawaki (1978) proposed to characterize image texture along the dimensions of coarseness, contrast, directionality, line-likeness, regularity and roughness. Coarseness corresponds to the "scale" or image resolution, contrast measure the amount of local intensity variation present in an image and directionality is a property due to both the basic shape of texture element and the placement rule used in creating the texture (ed. Castelli and Bergman 2002, p. 320).

In summary, these methods can be useful in image retrieval but are computationally expensive. There are a lot of projects going on for efficient texture based image retrieval, such as at UCSB and IBM.

5.3 Histogram-Based Color Image Retrieval

Histogram-based color image retrieval is the most common technique available and can be easily implemented. In this project, two histogram-based search methods are investigated and an attempt will be made to implement these methods in HSV color space.

5.3.1 Color Model

To understand this method, it is necessary to understand the HSV color model. This color model represent in terms of intensity values. The HSV stands for the Hue, Saturation, and Value based on the artists (Tint, Shade, and Tone).

The HSV color co-ordinate system and color model is shown below:

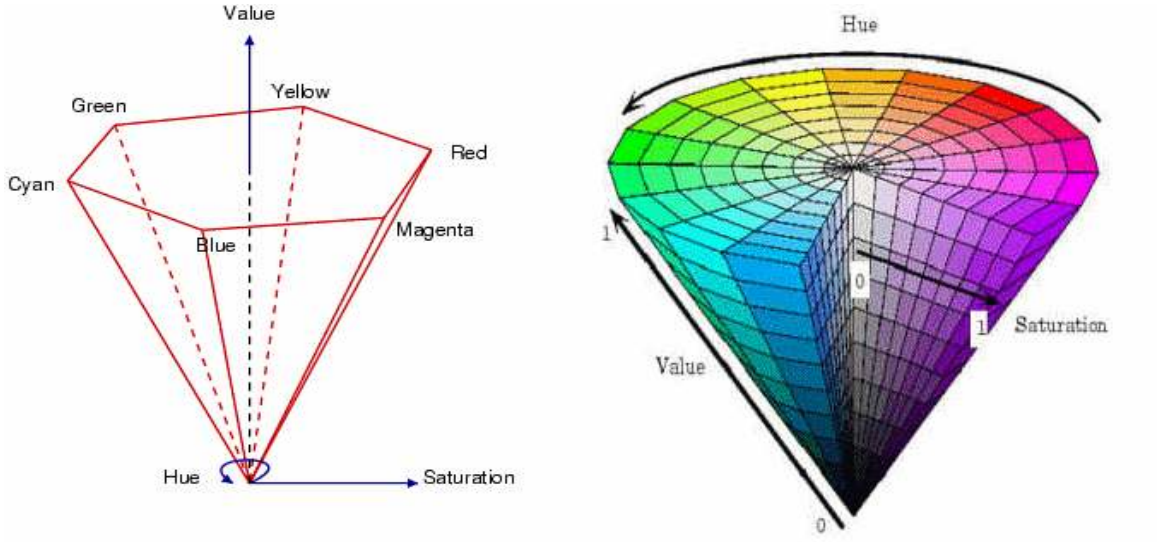

Fig 5.1: HSV co-ordinate system and color model
(Source: Jeong, 2001)

According to Jeong (2001), the hue and saturation components are intimately related to the way human eye perceives color resulting in image processing algorithms with physiological basis. As hue varies from 0 to 1, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values both at 0 and 1. As saturation varies from 0 to 1, the corresponding colors vary from unsaturated to fully saturated. As value, or brightness, varies from 0 to 1, the corresponding colors become increasingly brighter.

5.3.2 Histogram-Based Search

According to Jeong (2001), histogram based search usually follow this progression: (1) selection of a color space, (2) quantization of the color space, (3) computation of histograms and (4) derivation of the histogram distance function. Each of these steps may be crucial towards developing a successful algorithm.

These steps are described below:

1) In this step, HSV color space is chosen and all the database images and query image is taken in HSV color space. Since most of the machine read images in RGB color space, the conversion of RGB to HSV color space is done in this step. The conversion is done following these formulas:

$$H = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\},$$

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

$$V = \frac{1}{3}(R+G+B)$$

2) This step generally requires substantial quantization of the HSV color space. Using HSV color table, the color space is divided into eight different colors. These colors are: black, white, red, yellow, green, cyan, blue and magenta.

3) In this step, histograms of database images and query images are computed. An image histogram refers to the probability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three color channels. More formally, the color histogram is

defined by, $h_{A,B,C}(a,b,c) = N \cdot \mathbf{Pr}ob(A = a, B = b, C = c)$

where *A, B* and *C* represent the three-color channels (H, S, V) and *N* is the number of pixels in the image. Computationally, the color histogram is formed by counting the number of pixels for eight different colors.

4) This step describes several distance formulas for measuring the similarity of color histograms. Two distance formulas that have been used for image retrieval including histogram Euclidean distance and histogram intersection. The Euclidean distance between the color histograms ***h*** and ***g*** can be computed as $d^2(h,g) = \sum_A \sum_B \sum_C (h(a,b,c) - g(a,b,c))^2$ in this distance formula, there is only comparison between the identical bins in the respective histograms. Two different bins may represent perceptually similar colors but are not compared cross-wise. All bins contribute equally to the distance. The intersection of histograms ***h*** and ***g*** is given by: $d(h,g) = \frac{\sum_A \sum_B \sum_C \mathbf{min}(h(a,b,c), g(a,b,c))}{\mathbf{min}(|h|,|g|)}$ where $|\boldsymbol{h}|$ and $|\boldsymbol{g}|$ gives the magnitude of each histogram, which is equal to the number of samples. Colors that are not present in the user's query image do not contribute to the intersection distance. This reduces the contribution of background colors. The sum is normalized by the histogram with fewest samples.

Using these two distance methods, similarity is computed between query image and the database images. The image that returns the least distance assumed to be similar image and thus this method is implemented.

5.3.3 Overall Scheme

This method for image retrieval is quite is easy to implement and widely accepted. Though this method is not always accurate for image retrieval, it serves the idea of content-based approach.

The flow of this method is shown below:

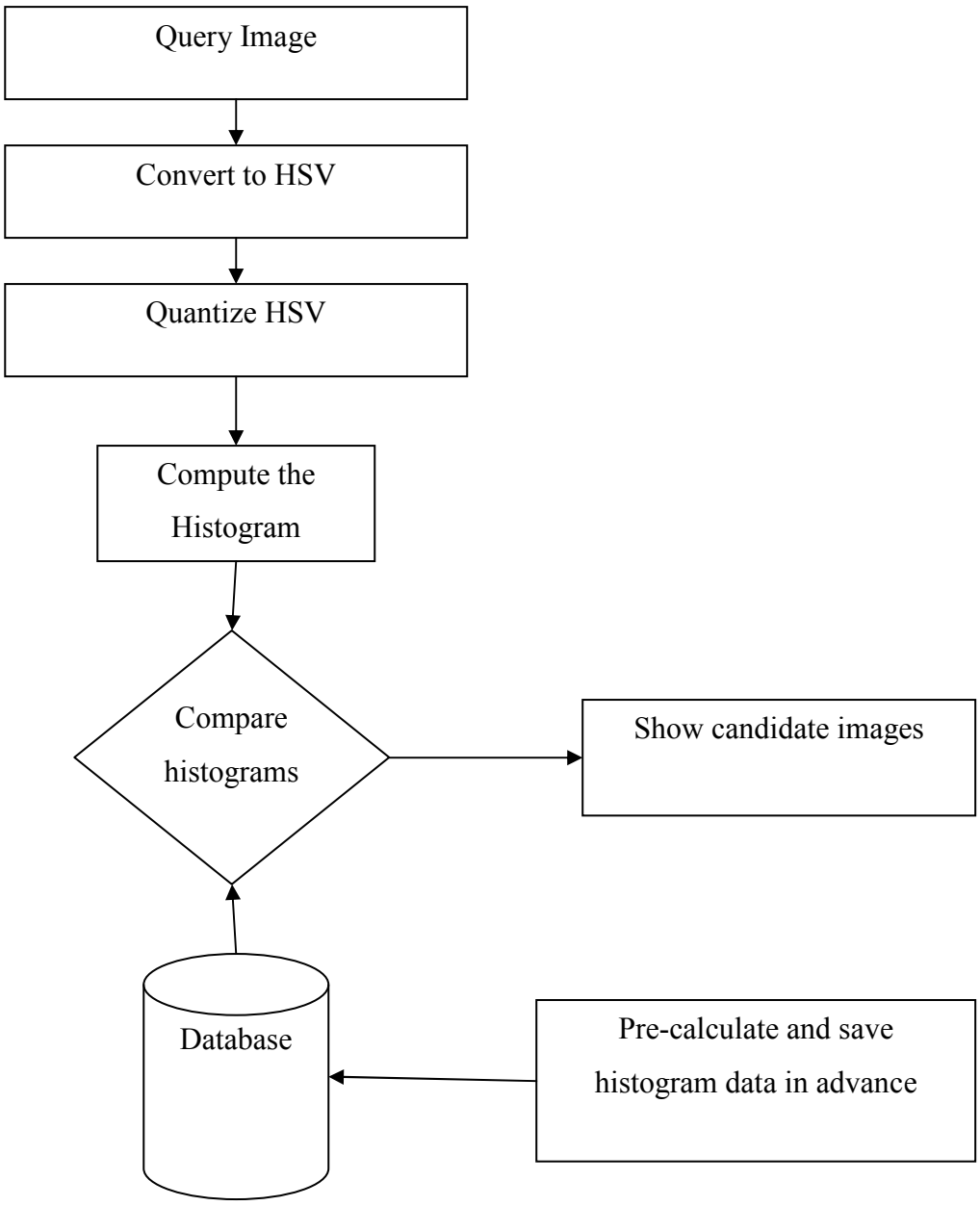

Fig 5.2: Flow of histogram-based search technique

In figure 5.2, it is shown that once a user inserts a query image, the rest of whole process is done automatically. However, the histogram data for all images in the database are computed and saved in database in advance so that only the image indexes and histogram data can be used to compare the query image with images in database.

5.4 Chapter Summary

The image indexing and retrieval techniques discussed in this chapter are normally called low-level, content-based image retrieval techniques. The common characteristics of these techniques are, they derive a feature vector for each image and the similarity between two images is calculated based on their corresponding vectors. Although there are many image retrieval techniques available, no single technique proved to be the perfect one. Hence, this chapter discusses different approaches to image retrieval. In addition, Histogram-based retrieval is briefly described, as this project aims to implement the technique.

Chapter 6

Video Indexing and Retrieval

6.1 Introduction

Recent advances in computing, communication, and data storage have led to an increasing number of large digital libraries publicly available on the internet. In addition to alphanumeric data, other modalities, including video play an important role in these libraries. Ordinary techniques will not retrieve required information from the enormous mass of data stored in digital video libraries. Hence, content-based video retrieval techniques are necessary to deal with this situation.

This chapter provides an overview of possible approaches to video retrieval. The fundamental approaches to video retrieval are identified and listed below:

- Shot-Based Video Retrieval
- Motion Information-Based Video Retrieval
- Object-Based Video Retrieval

In the following sections, each of these approaches will be described briefly.

6.2 Possible Approaches to Video Retrieval

Content-based Video Retrieval seems like a natural extension (or merge) of Content-based Image Retrieval and Content-Based Audio Retrieval. However, there are a number of factors that are ignored when dealing with images, which should be dealt with when using videos. These factors are primarily related to the temporal information available from a video document. While these factors may complicate the querying system, they also may help in characterizing useful information for the querying.

Content-based video retrieval is still in its early development stage. Since this area is fairly recent, there is no universally accepted retrieval technique available. Many researchers proposed many different techniques extending current image and audio retrieval techniques. But only a few methods have been proved to be worth researching. Lu (1999) discussed some methods for video retrieval based on low-level features and they are described below:

6.2.1 Shot-Based Video Retrieval

In this technique, internal video property named video shot is used for retrieval. Video is normally made of several logical units or segments. These segments are called video shots. According to Lu (1999), a shot is a short sequence of contiguous frames having one or more of the following feature:

- The frames illustrate the same scene.
- The frames indicate a single camera operation.
- The frames contain a unique event such as presence of an object.

For example, in a shopping network video, each shopping item can corresponds to a shot.

According to Lu (1999), the main steps for shot-based video retrieval are:

Segmentation of Video: In this step, the video is segmented into shots. This step is named as video temporal segmentation, partition or shot detection.

Indexing shots: In the second step, each shot is indexed. The general approach used in this step is to identify key frames for each shot and use any image indexing method to index this frames.

Similarity measurement: The third step is to compute similarity measurement between query and video shots and retrieve shots that are similar. This is achieved using image retrieval techniques based on feature vectors.

An example is shown below for shot-based and object-based video retrieval system. This system was created at Centre for Digital Video Processing in Ireland. Figure 6.1 depicts object based query process:

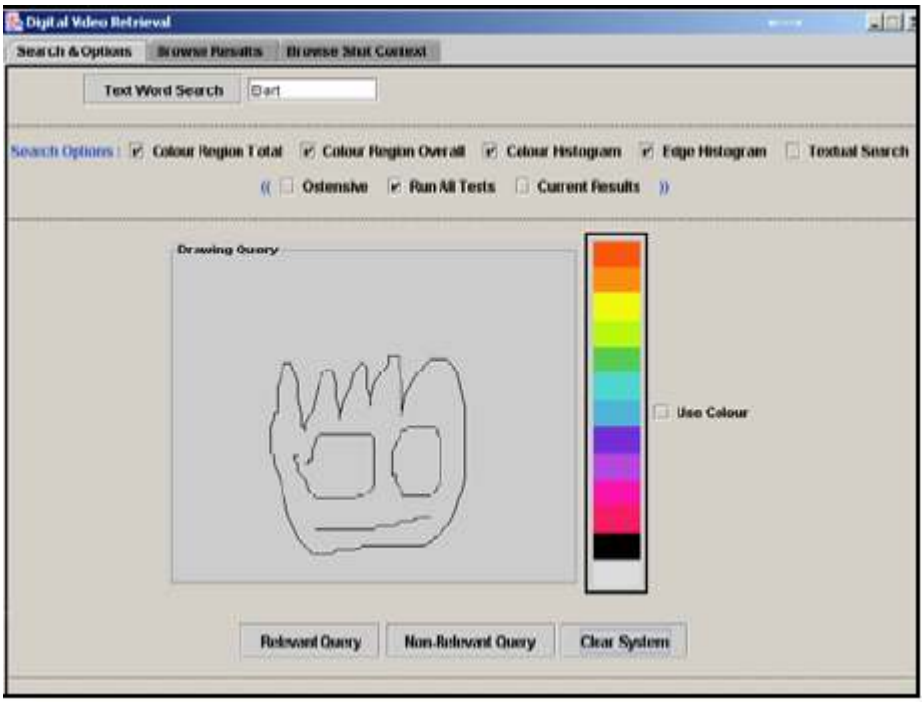

Fig 6.1: Interface for searching video shots
(Source: Browne & Smeaton 2004, p. 1087)

Figure 6.2 shows object based retrieval as shots and figure 6.3 shows the capability of the system to browse through shots.

Fig 6.2: shot based retrieval

(Source: Browne & Smeaton 2004, p. 1088)

Fig 6.3: Browse shot control

(Source: Browne & Smeaton 2004, p. 1088)

6.2.2 Motion Information-Based Video Retrieval

Motion information of a video can be derived from optical flow or motion vectors. Hampapur (1997) suggested the following parameters for motion indexing:

Motion Content: It measures the total amount of motion within a video. It basically measures the action content of the video. For example, a news video has very less motion content compare to a car crash video.

Motion Uniformity: This parameter defines the smoothness of the motion within a video as a function of time. For example, a smooth panning shot has high value of motion smoothness, while a video with a staggered pan has a low value.

Motion Panning: This parameter restrains the panning motion. Panning motion means the motion that is caused by motion of camera from left to right or right to left. A smooth pan shot scores higher than a zoom shot.

Motion Titling: It measures the vertical motion component of the motion within a video sequence. Panning shots have a lower value than a video with large amount of vertical motion.

These motion parameters are associated with an entire video stream or a video shot and can be used for video retrieval. Using these parameters as feature vectors video retrieval method can be devised.

6.2.3 Object-Based Video Retrieval

In a video, any scene is a complex combination of different objects. Objects basically define the scene by their location, physical qualities and interaction with each other. Object-Based video retrieval proposes to find a way to distinguish individual objects through out the video sequence and index the videos according to objects found (Lu

1999, p. 189). During the retrieval process, the indexes of database can capture the changes in content throughout the sequence of the query video.

Object segmentation and identification is normally complicated and difficult in still image processing. But in the case of video sequence, it is easy to implement. In a video, an object usually moves as a whole. So it is possible to group pixels that move together into an object. Thus object segmentation can be implemented. Subsequently, these objects can be used as the basis for video indexing and retrieval.

6.3 Chapter Summary

Content-based video retrieval is a complex process, because there is so much variety in video imagery. In this chapter, three approaches to video retrieval are discussed. However, these approaches only handle the low-level features. High-level features such as temporal events and interaction between objects within a video are still hard to identify and extract. Thus, more research is required to achieve video retrieval based on Hi-level features. In addition, some screen shots of a demo video retrieval system are shown to provide understanding of shot-based video retrieval.

Chapter 7

Performance of Selected Search Techniques

7.1 Introduction

In Chapter 4 and 5, several methods for audio and image retrieval were discussed. These methods were briefly described and it was mentioned that an attempt will be made to implement those techniques. In this chapter, the results of the implemented techniques will be described. In addition, some related issues are identified and listed below:

- Computational efficiency.
- Discussion of results.

In the following sections, each of these issues will be discussed in greater detail.

7.2 Results for Audio Retrieval

The following methods were described in chapter 4:

- LPC analysis
- Filter-Bank analysis

These methods are implemented using MATLAB to retrieve audio speech files and the codes can be found in appendix B. Test files are downloaded from:

<http:://lcavwww.epfl.ch/~minhdo/asr_project/data/data.zip#here>

There are two folders, TRAIN and TEST, each contains 8 files, named: S1.WAV, S2.WAV… S8.WAV; each is labeled after the ID of the speaker. In these files, each speaker says the word "zero". These files are recorded in Microsoft WAV format. The goal is to train a voice model (or more specific, a VQ codebook using LPC & Filter-bank) for each speaker S1 - S8 using the corresponding sound file in the TRAIN folder. After this training step, the system would have knowledge of the voice characteristic of each (known) speaker. For convenience, the wave files at TEST & TRAIN directory are transferred to the same directory and test files are renamed like S11.WAV, S12.WAV…S18.WAV to represent the same speaker from S1.WAV to S8.WAV. Then the testing is carried out to identify the (assumed unknown) speaker of each sound file.

At first the system shows the following:

Fig 7.1: The system interface

After choosing the audio retrieval option, the following window appears:

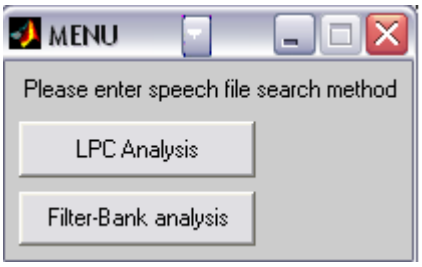

Fig 7.2: Audio search options

Then LPC analysis is selected to search for a file and the system asks for an example file to search the database:

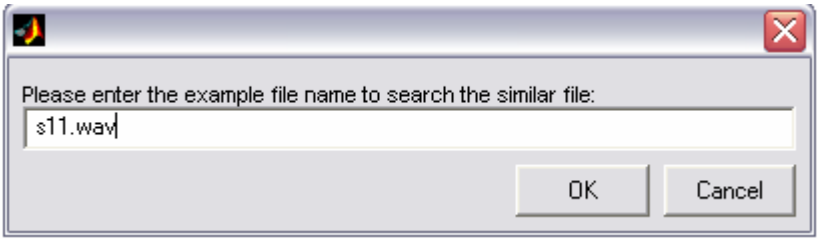

Fig 7.3: Inserting an example file to search the audio database

The system represents the result:

Fig 7.4: Result of audio search

Then the method mentioned above is repeated for 8 different speech files using LPC analysis and the following results are found:

Query File	**Retrieved File**
s11.wav	s1.wav
s12.wav	s2.wav
s13.wav	s3.wav
s14.wav	s4.wav
s15.wav	s8.wav
s16.wav	s6.wav
s17.wav	s8.wav
s18.wav	s18.wav

Table 7.1: Results of audio retrieval using LPC analysis

Then, the test for Filter-Bank analysis is carried out. Table 7.2 shows the results obtained:

Query File	**Retrieved File**
s11.wav	s1.wav
s12.wav	s2.wav
s13.wav	s6.wav
s14.wav	s4.wav
s15.wav	s5.wav
s16.wav	s3.wav
s17.wav	s5.wav
s18.wav	s7.wav

Table 7.2: Results of audio retrieval using Filter-Bank analysis

In summary, this section provides the results of LPC analysis and filter-bank analysis for speech file recognition. Basically, these methods tried to recognize a speaker based on the training data. As can be seen from the result tables, the system does not always recognize the speakers correctly.

7.2 Results for Image Retrieval

In chapter 5, it was mentioned that implementation of color based histogram search technique is one of the task of this project. Therefore, this technique is implemented by using MATLAB and the codes can be found in appendix B. Test files are downloaded from:

<http://wang.ist.psu.edu/docs/related/>

There are one thousand color images and for this testing only thirty images are considered for computational convenience. In these thirty images, there are three similar types of images. They are:

- Ten images of flowers
- Ten images of buses
- Ten images of mountains

Then six other images are considered as test files and each two of them belongs to one category of images. The retrieval process is shown below:

At first, the system shows:

Fig 7.5: The system interface

Then, Image retrieval is chosen and the system asks for input image:

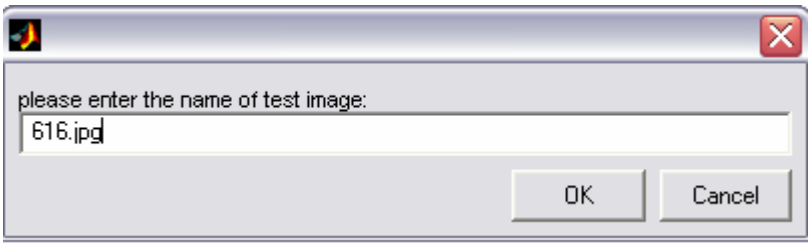

Fig 7.6: Inserting example file to retrieve similar file

The first input image is:

Fig 7.7: The first input image (616.jpg)

The results of the system are:

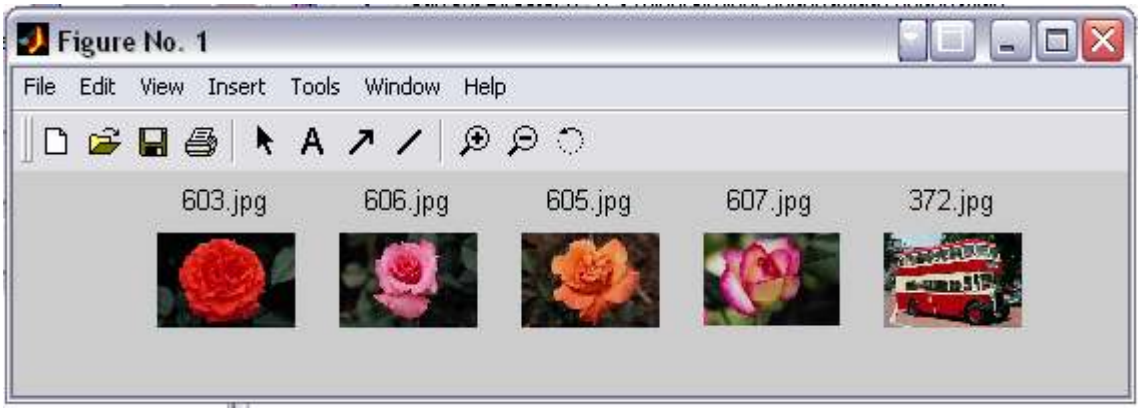

Fig 7.8: Best 5 retrieval using Histogram Intersection for 616.jpg

Fig 7.9: Best 5 retrieval using Histogram Euclidian method for 616.jpg

620.jpg is used for another test:

Fig 7.10: The second input image (620.jpg)

The results are:

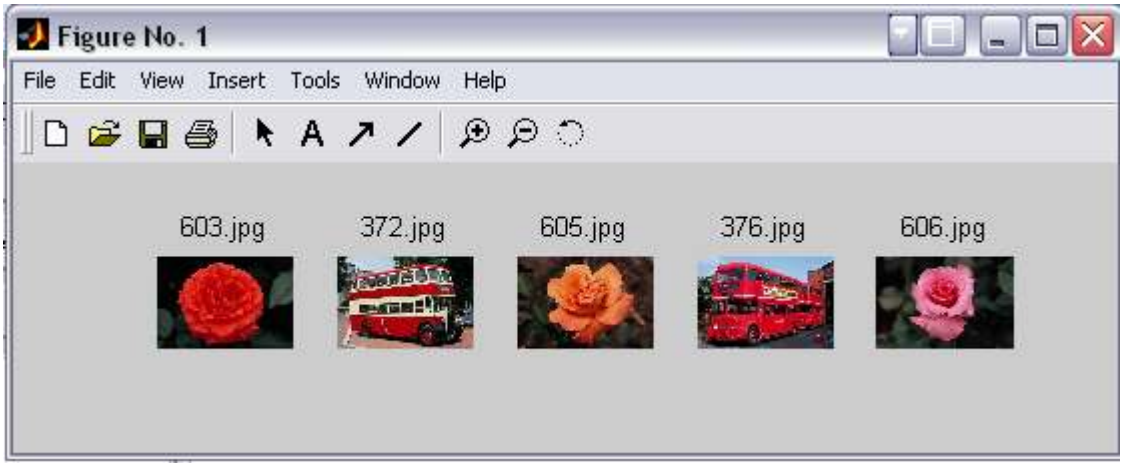

Fig 7.11: Best 5 retrieval using Histogram Intersection for 620.jpg

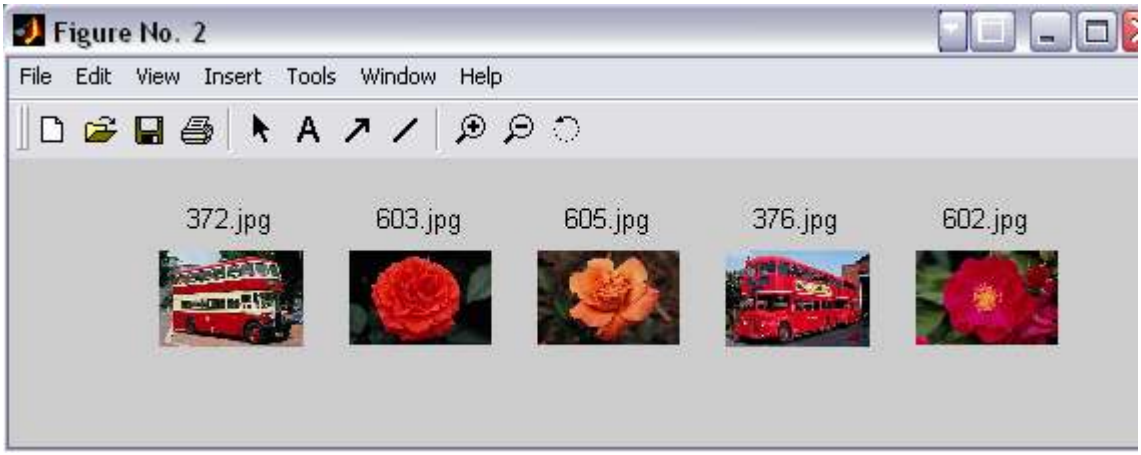

Fig 7.12: Best 5 retrieval using Histogram Euclidian method for 620.jpg

The next test image is 810.jpg:

Fig 7.13: The third input image (810.jpg)

The results are:

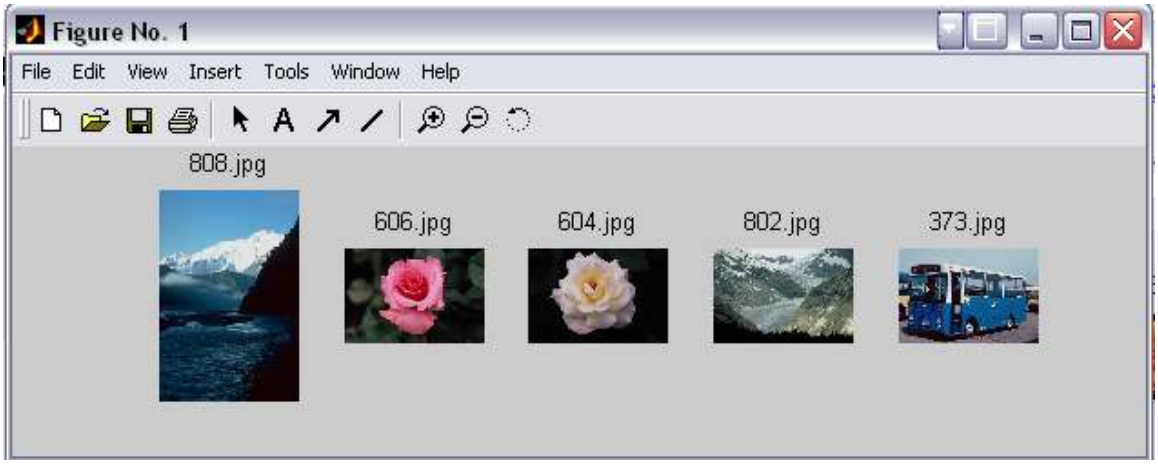

Fig 7.14: Best 5 retrieval using Histogram Intersection for 810.jpg

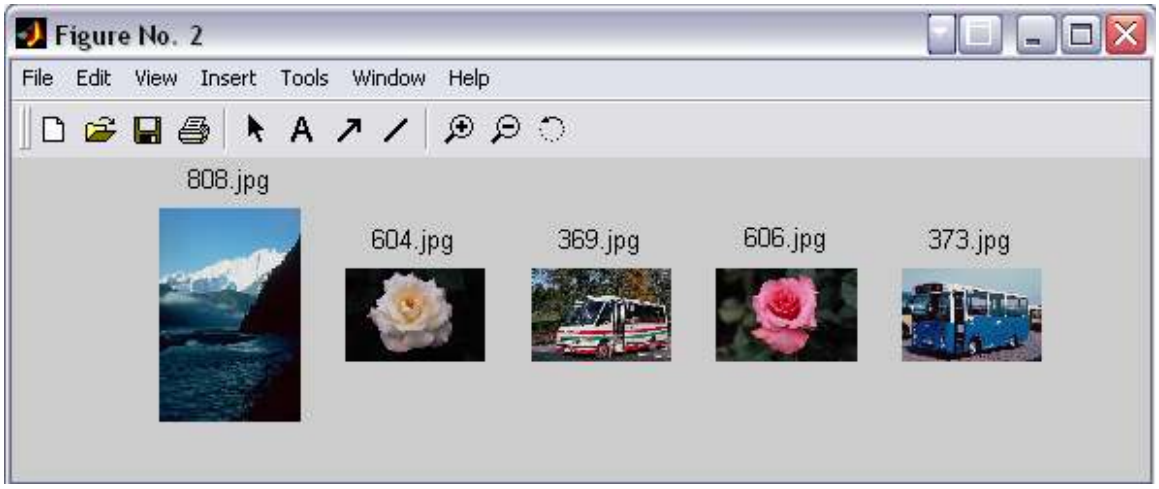

Fig 7.15: Best 5 retrieval using Histogram Euclidian method for 810.jpg

The next test image is 811.jpg:

Fig 7.16: The fourth input image (811.jpg)

The results are:

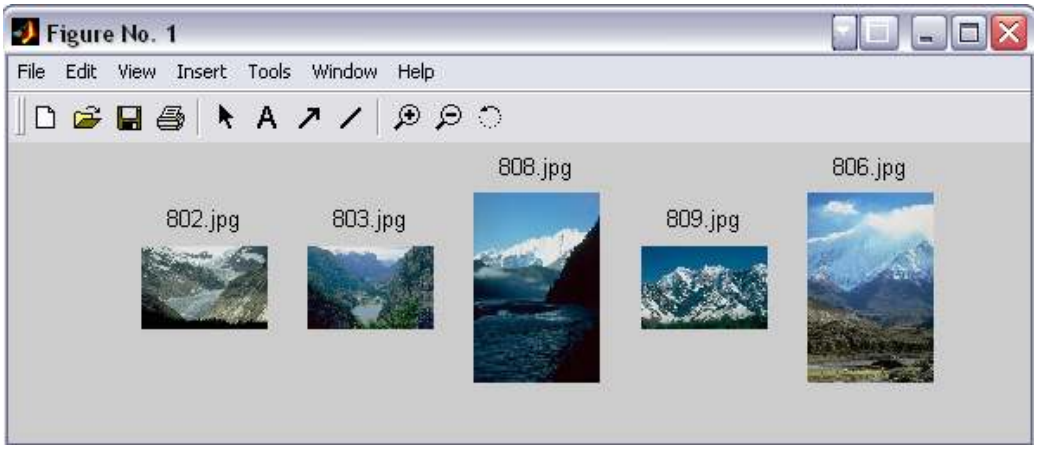

Fig 7.17: Best 5 retrieval using Histogram Intersection for 811.jpg

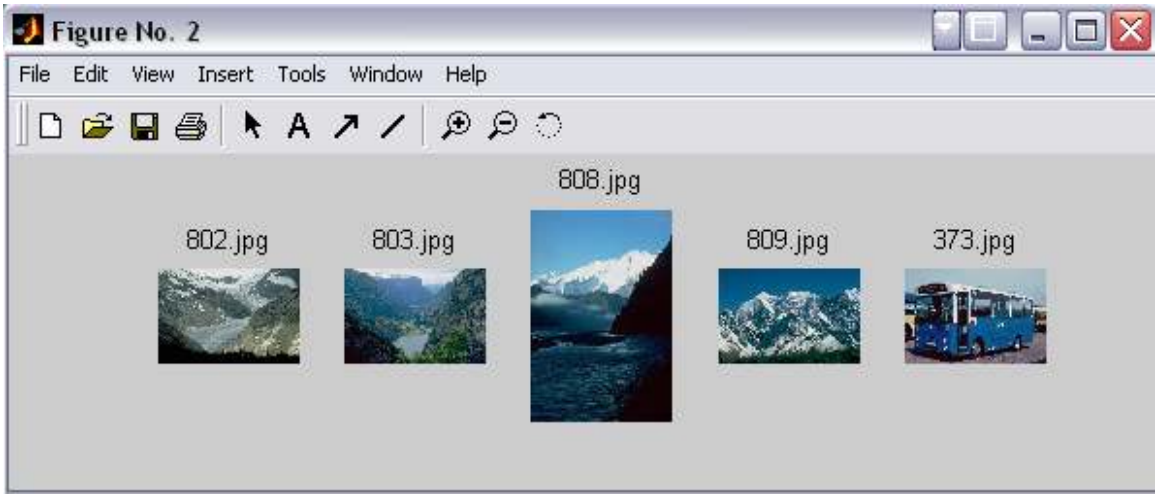

Fig 7.18: Best 5 retrieval using Histogram Euclidian method for 811.jpg

The next test image is 378.jpg:

Fig 7.19: The fifth input image (378.jpg)

The results are:

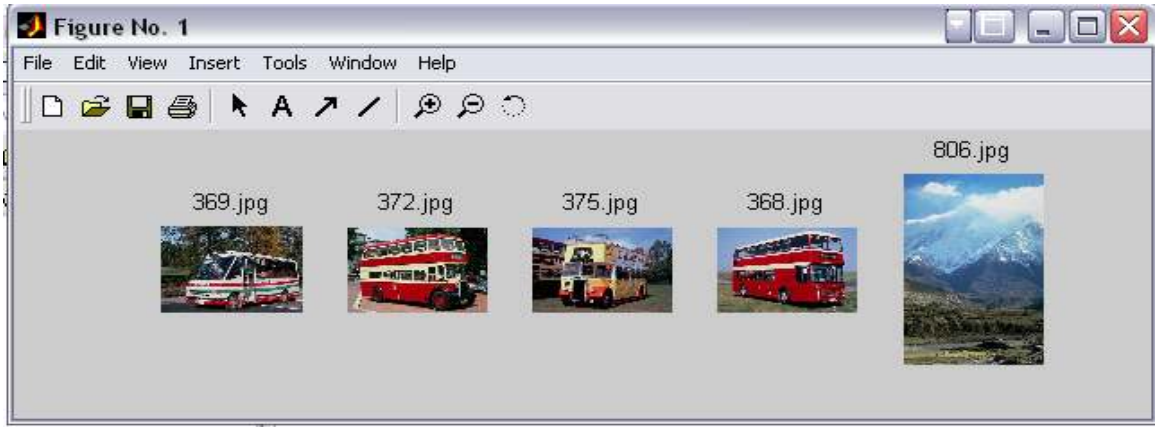

Fig 7.20: Best 5 retrieval using Histogram Intersection for 378.jpg

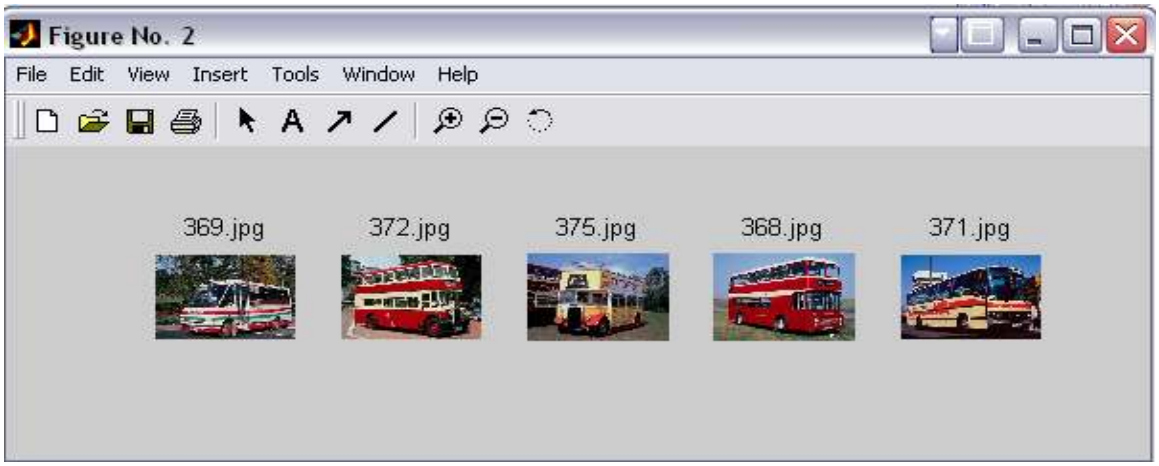

Fig 7.21: Best 5 retrieval using Histogram Euclidian method 378.jpg

The last test image is 379.jpg:

Fig 7.22: The sixth input image (379.jpg)

The results are:

Fig 7.23: Best 5 retrieval using Histogram Intersection for 379.jpg

Fig 7.24: Best 5 retrieval using Histogram Euclidian method for 379.jpg

7.4 Computational Efficiency

In this project, two methods are implemented for audio retrieval. These two methods are relatively straightforward and easy to implement. Therefore, the computation time for indexing speech database and calculating distances between the query and database items are reasonable. Since for speech database only eight speech files are used, it does not take long time to index the training data. In addition, histogram-based image retrieval technique is also easy to implement and it took reasonable amount of time to retrieve similar images. Although the system is fast for retrieving images, it took about thirty seconds to index those thirty database images.

In summary, the implemented three techniques are reasonably efficient in the context of computing. Though the system is computationally efficient, it takes a little while for indexing database images.

7.5 Discussion of Results

The results obtained by testing the implemented techniques are pretty reasonable. The discussion for audio and image retrieval is described below:

7.5.1 Audio Retrieval

For audio retrieval two methods were used and the analysis of the results:

- The LPC analysis recognized 6 speakers correctly and 2 speakers incorrectly. So, the success rate of this method is 75%.
- The Filter-Bank analysis recognized 4 speakers correctly and 4 speakers incorrectly. So, the success rate is 50%.

7.5.2 Image Retrieval

For image retrieval, a benchmark is intended to use to check retrieval effectiveness. In this regard, Jeong (2001) discussed about recall and precision. Recall indicates the proportion of relevant images in the database that are retrieved in response to a query. Precision measures the efficiency with which the relevant items are returned among the best matches. To describe precisely, let *A* be the set of relevant items, let *B* the set of retrieved items and *a, b, c* and *d* are given in figure 7.25

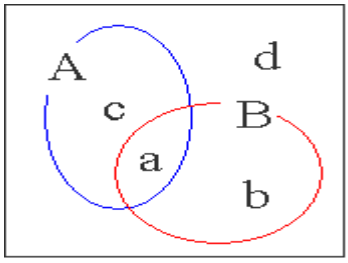

Fig 7.25: Sets for explaining retrieval effectiveness

(Source: Jeong, 2001)

In figure 7.25, ***a*** stands for 'retrieved relevant' images, *b* for 'retrieved irrelevant' images, *c* for 'unretrieved relevant' images and *d* for 'unretrieved irrelevant' images.

Then according to Smith and Chang (1996), recall and precision are defined as the following conditional probabilities:

$$recall = P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{a}{a+c},$$

$$precision = P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{a}{a+b}$$

With these conditions, image retrieval is said to be more effective if precision values are higher at the same recall values. Hence, the recall and precision values are calculated & the results are given below:

Query Image	Histogram Intersection method		Histogram Euclidean method	
	Recall	Precision	Recall	Precision
616.jpg	0.4	0.8	0.4	0.8
620.jpg	0.3	0.3	0.3	0.6
810.jpg	0.2	0.4	0.1	0.2
811.jpg	0.5	1	0.4	0.8
378.jpg	0.4	0.8	0.5	1
379.jpg	0.4	0.8	0.4	0.8

Table 7.3: Retrieval effectiveness

By investigating table 7.3, it can be concluded that this color based image retrieval technique is effective for 616.jpg, 811.jpg and 378.jpg as their precision values are higher in both methods. Although these two methods retrieved poorly for other files, the systems performance is reasonable.

7.6 Chapter Summary

In this chapter, the results of two histogram-based search techniques and two methods for audio speech file retrieval are discussed. In addition, computational efficiency and retrieval effectiveness of these methods are also investigated. This chapter provides a good understanding about internal concept of this project. Although some results of the implemented methods of this chapter are not always accurate, it provides a good research direction in this area.

Chapter 8

Conclusion & Further Work

8.1 Accomplishment of Aims

The following objectives have been addressed:

Overview of Multimedia Database architecture, design issues and multiple media: Chapter 2 presented a brief description of multimedia database architecture and multiple types of media. It also described current issues related to multimedia database management. Different design issues relating multimedia database management are extensively researched in order to get a better understanding of multimedia database search techniques.

Investigation of commonly used Multimedia Database search techniques: Chapter 3 depicted a standard model for multimedia retrieval and their expected capabilities. Commonly used search techniques for audio, image and video files are investigated to satisfy the overall project objective.

Investigation of Audio Retrieval: Chapter 4 described about audio properties and possible approaches for audio classification and retrieval. Among these approaches LPC analysis and Filer-Bank analysis are briefly investigated to understand the idea of speech file retrieval.

Investigation of Image Retrieval: Chapter 5 represents discussion of different image retrieval techniques. Histogram-Based color image retrieval is briefly examined to gain an understanding of content-based image retrieval.

Investigation of Video Retrieval: Chapter 6 explores possible approaches to video retrieval. A demo video retrieval devised at Centre for Digital Video Processing (Ireland) is investigated to gain an understanding about shot-based video retrieval.

Implementation of audio and image retrieval techniques: Chapter 4 briefly described LPC method and Filter-Bank analysis method for audio retrieval and Chapter 5 described histogram-based color image retrieval. These methods are implemented using MATLAB and their results are discussed in Chapter 7. The results obtained show that LPC method and histogram intersection method performs better compare to filter-bank analysis and histogram Euclidean method respectively.

8.2 Further Work

There are several areas for possible further research and development. These areas include:

- Further investigation to identify possible approaches for integrated audio, image and video retrieval using different compressed data.
- Further research to derive an integrated approach to video retrieval using spatial and temporal analysis of video data.
- Further study to develop a user-friendly user interface to search multimedia documents.
- Enhancement of audio, image and video indexing as they take a considerable amount of time to pre-calculate the feature vectors and initiate the database.

References

Lu, G, 1999, *Multimedia Database Management Systems,* Artech House, Boston.

Li, ZN & Drew, MS, 2004, *Fundamentals of Multimedia*, Prentice Hall, New Jersey.

Rabiner, L & Juang, BH, 1993, *Fundamentals of Speech Recognition*, Prentice Hall, New Jersey.

Castelli, V & Bergman, LD(ed.) 2002, *Image Databases: Search and Retrieval of Digital Imagery,* John Wiley & Sons, New York.

Bimbo, AD 1999, *Visual Information Retrieval,* Morgan Kaufmann Publishers, San Francisco.

Bohm, C, Berchtold, S & Keim, DA, 'Searching in High-Dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases', *ACM Computing Surveys*, Vol. 33, No. 3, September 2001, pp. 322–373.

Grosky, WI 1997, 'Managing Multimedia Information in Database Systems', *Communications of the ACM,* vol. 40, no.12, p. 73 – 80, viewed 10th March 2005, ACM.

Jain, AJ & Vailaya, A, 'Image Retrieval Using Color and Shape', *Proceeding of Second Asian Conference on Computer Vision,* Dec. 5-8 1995, Singapore, p. 529-533.

Julesz, B, 'Experiments in the Visual Perception of Texture', *Sci. Am.* 232(4), p. 2-11, 1975.

Tamura, H, Mori, S & Yamawaki T, 'Texture Features Corresponding to Visual Perception', IEEE *Trans. Sys. Man, Cybernatics 8(6)*, p. 460-473, 1978.

Wan, X & C. J. Kuo, 'Color Distribution Analysis and Quantization for Image Retrieval' *Proceedings of Conference on Storage and Retrieval for Image and Video Database 4*, Feb. 1-2, 1996, San Jose California, SPIE Vol. 2670, pp. 8-16.

Hampapur, A, 'Virage Video Engine', *Proceeding of the Conference on Storage and Retrieval of Image and Video Databases,* San Jose, CA, Feb. 13-14 1997, SPIE Proceeding Series, Vol. 3022, p. 188-198.

Browne, P & Smeaton, PB, 'Video Information Retrieval Using Objects & Ostensive Relevance Feedback', *ACM Symposium on Applied Computing*, Cyprus, March 14-17 2004, SAC'04, p. 1084-1090.

Qasem, M 1997, 'Vector Quantization', viewed 15th April 2005, < http://www.geocities.com/mohamedqasem/ vectorquantization/vq.html>.

Jeong, S 2001, 'Histogram Based Color Image Retrieval', viewed 20th May, 2005, < http://scien.stanford.edu/class/psych221/projects/02/sojeong/>.

Smith, JR & Chang, SF, 'Tools and techniques for Color image retrieval', *Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV*, volume 2670, San Jose, CA, February 1996. IS&T/SPIE.

C-BIRD Interface, 1999, C-BIRD, US, viewed 31 March 2005, <http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/images/cbird1.jpg>.

QBIC Interface, 1997, QBIC, US, viewed 4th April 2005, <http://www.cse.unsw.edu.au/~jas/talks/curveix1/pic/qbic1.jpg>.

Neon Signs Interface, 2000, Neon Signs, viewed 8th April 2005, <http://www-128.ibm.com/developerworks/db2/library/techarticle/0202cox/images/part2fig4.gif>.

Codeword in 2-D Space, 2000, Vector Quantization, viewed 10th May 2005, < http://www.geocities.com/mohamedqasem/vectorquantization/vq.html>.

Soundfisher Interface, 1999, SOUNDFISHER, viewed 26th May 2005, < http://www.soundfisher.com/html/tour1.html>.

Appendix A

Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

ENG 4111/4112 Research Project
PROJECT SPECIFICATION

FOR:	Nizam Khan
TOPIC:	Multimedia Database Search Techniques
SUPERVISOR:	Dr. John Leis
PROJECT AIM:	The project aims to investigate possible approaches to search documents from a multimedia database and to prototype some systems to test their performance.

PROGRAMME: **Issue A, 4th April 2005**

1. Research background information about different media types and their storage and compression techniques.
2. Research issues related to multimedia database design: integrated approach to multimedia indexing and retrieval, feature extraction, content representation and similarity measurement.
3. Investigate common approaches for video indexing, retrieval and related issues.
4. Investigate available multimedia databases including various media types, format and availability.
5. Investigate audio search and retrieval algorithms. Research issues related to audio document and their properties and features. Investigate the techniques for: speech recognition and retrieval, music indexing and retrieval.
6. Investigate image search and retrieval algorithms. Research different approaches to image indexing and retrieval such as, text based image retrieval, color based image indexing and retrieval, image retrieval based on shape, image retrieval based on texture and image retrieval based on compressed image data.

7. Design an algorithm and implement a program in MATLAB to retrieve an audio document from a multimedia database.

As time permits:

8. Design an algorithm and implement a program in MATLAB to retrieve an image document from a multimedia database.

AGREED: Nizam Khan (student) _____ (supervisor)

DATE: 04/04/2005

Appendix B

Source Codes

```
% main.m
% Main file that implements multimedia database
% Written by: Nizam Khan, 2005.

K = 0;
warning off;

%creating image and audio database
[histmat,pixels] = image_database;
CodeBook1 = audio_database_lpc;
CodeBook2 = audio_database_filterbank;

while (K ~= 3)
    K = menu('Please Enter search options','Audio
Retrieval', 'Image Retrieval','Exit');
    close all;
    if ( K ~= 3)
        if (K == 1)
            K_K = menu('Please enter speech file search
method','LPC Analysis','Filter-Bank analysis');
            if (K_K == 1)
                %fucntion file to do LPC analysis
                audio_recog_lpc(CodeBook1);
            elseif (K_K == 2)
                %function file to do filterbank analysis
                audio_recog_filterbank(CodeBook2);
            end
        elseif (K == 2)
               %function file to histogram-based search
               image_recog(histmat,pixels);
        end
    else
```

```
        msgbox('Exiting the program.........');
    end
end
```

```
% audio_database_filterbank.m
% Function file to train the system with audio data for
% Filterbank analysis
% written by Nizam khan, 2005

function CodeBook = audio_database_filterbank()
CodeBook = zeros(64,28);

[soundvec sr nbits] = wavread('s1.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(1:8,:) = center;

[soundvec sr nbits] = wavread('s2.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(9:16,:) = center;

[soundvec sr nbits] = wavread('s3.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(17:24,:) = center;

[soundvec sr nbits] = wavread('s4.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(25:32,:) = center;

[soundvec sr nbits] = wavread('s5.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(33:40,:) = center;
```

```
[soundvec sr nbits] = wavread('s6.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(41:48,:) = center;

[soundvec sr nbits] = wavread('s7.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(49:56,:) = center;

[soundvec sr nbits] = wavread('s8.wav');
e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
CodeBook(57:64,:) = center;
```

```
% audio_database_lpc.m
% Fucntion file that creates audio data base for LPC
% method
% Written by Nizam khan, 2005

function CodeBook = audio_database_lpc()

CodeBook = zeros(32,11);

[soundvec sr nbits] = wavread('s1.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(1:4,:) = center;
[soundvec sr nbits] = wavread('s2.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(5:8,:) = center;
[soundvec sr nbits] = wavread('s3.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(9:12,:) = center;
[soundvec sr nbits] = wavread('s4.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
```

```
CodeBook(13:16,:) = center;
[soundvec sr nbits] = wavread('s5.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(17:20,:) = center;
[soundvec sr nbits] = wavread('s6.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(21:24,:) = center;
[soundvec sr nbits] = wavread('s7.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(25:28,:) = center;
[soundvec sr nbits] = wavread('s8.wav');
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';
[center, U] = vqLBG_bb(aCoeff,4);
CodeBook(29:32,:) = center;
```

```
% audio_recog_filterbank.m
% Function file that uses Filter-Bank analysis method for
% speaker recognition
% Written by: Nizam Khan

function audio_recog_filterbank(CodeBook)

%Getting query example file
aud = inputdlg('Please enter the example file name to
search the similar file:');

audiofile = char(aud);
[soundvec sr nbits] = wavread(audiofile);

e = auditoryfbank(soundvec,sr);
[center, U] = vqLBG_bb(e,8);
dmin = realmax;
n = 1;
while n < 58
    dist = (center - CodeBook(n:n+7,:)).^2;
    distortion = sum(sum(dist));
    if distortion < dmin
       dmin = distortion;
       index = n;
    end
n=n+8;
end
audioshow_filterbank(index);
```

```
% audio_recog_lpc.m
% Function file that uses LPC method for speaker
% recognition.
% Written by: Nizam Khan

function audio_recog_lpc(CodeBook)

%Getting query example file
aud = inputdlg('Please enter the example file name to
search the similar file:');

audiofile = char(aud);
[soundvec sr nbits] = wavread(audiofile);

%Applying LPC analysis
[aCoeff,resid,pitch,G,parcor,stream] =
proclpc(soundvec,sr,10);
aCoeff = aCoeff';

%Applying Vector Quantization
[center, U] = vqLBG_bb(aCoeff,4);
%Calculating distance using Euclidean distance
dmin = realmax;
n = 1;
while n <30
      dist = (center - CodeBook(n:n+3,:)).^2;
      distortion = sum(sum(dist));
      if distortion < dmin
         dmin = distortion;
         index = n;
      end
n=n+4;
```

```
end
%Retrieving similar audio file
audioshow_lpc(index);
```

```
% audioshow_filterbank.m
% Function file that retrieves similar audio file
% written by: Nizam Khan, 2005

function audioshow_filterbank(ind)

switch (ind)

     case 1
        msgbox('s1.wav','The Similar file matches
according to filterbank analysis:');
     case 9
        msgbox('s2.wav','The Similar file matches
according to filterbank analysis:');
     case 17
        msgbox('s3.wav','The Similar file matches
according to filterbank analysis:');
     case 25
         msgbox('s4.wav','The Similar file matches
according to filterbank analysis:');
     case 33
         msgbox('s5.wav','The Similar file matches
according to filterbank analysis:');
     case 41
         msgbox('s6.wav','The Similar file matches
according to filterbank analysis:');
     case 49
         msgbox('s7.wav','The Similar file matches
according to filterbank analysis:');
     case 57
         msgbox('s8.wav','The Similar file matches
according to filterbank analysis:');
```

```
end
```

```
% audioshow_lpc.m
% Function file that retrieves similar audio file
% Written by: Nizam Khan, 2005

function audioshow_lpc(ind)

switch (ind)
   case 1
      msgbox('s1.wav','The Similar file matches according
to filterbank analysis:');
   case 5
      msgbox('s2.wav','The Similar file matches according
to filterbank analysis:');
   case 9
      msgbox('s3.wav','The Similar file matches according
to filterbank analysis:');
   case 13
      msgbox('s4.wav','The Similar file matches according
to filterbank analysis:');
   case 17
      msgbox('s5.wav','The Similar file matches according
to filterbank analysis:');
   case 21
      msgbox('s6.wav','The Similar file matches according
to filterbank analysis:');
   case 25
      msgbox('s7.wav','The Similar file matches according
to filterbank analysis:');
   case 29
      msgbox('s8.wav','The Similar file matches according
to filterbank analysis:');
end
```

```
% auditoryfbank.m

function e=auditoryfbank(x,fs)

%   AUDITORYFBANK performs an auditory filterbank analysis
%   on a signal
%   E=AUDITORYFBANK(X,FS) passes signal X, sampled at FS
%   samples/sec
%   through a 32-channel auditory filterbank, smoothing
%   and downsampling energies to 100 frames per second.
%   get output energy sample points
%   Originally coded at:
%http://www.phon.ucl.ac.uk/courses/spsci/matlab/lect9.html
%   Modified by: Nizam Khan, 2005

% force to be a column
x=reshape(x,[length(x) 1]);

% sampling indices (100 %frame/sec)
eidx=fs/100:fs/100:length(x);

% number of output
ne=length(eidx);

%energy frames
%these are the filter cut-offs
cuts=[180 300 420 540 660 780 900 1075 1225 1375 1525 1700
1900 2100 ...
    2300 2550 2850 3150 3475 4000 4250 4450 4750 5000 5300
5550 5750 5950 6200];

% number of filters
```

```
nf=length(cuts)-1;

% initialise the output energy table
e=zeros(ne,nf);

% build a smoothing filter
% low-pass at half output frame rate
[sb,sa]=butter(4,2*50/fs);

% for each channel in turn
for i=1:nf
    % build the band-pass filter
    [b,a]=butter(4,[2*cuts(i)/fs 2*cuts(i+1)/fs]);
    % filter the signal
    y=filter(b,a,x);
    % rectify and smooth
    sy=filter(sb,sa,abs(y));
    % sample
    % force to be > 0
    sy=max(sy,eps);
    e(:,i)=(sy(eidx));
end
```

```
% colorhistf.m
% Computes histogram of colors present in picture.  Each
% pixel's color is quanitized to be either black, white,
% red, yellow, green, blue, cyan, or magenta.  These
% quanitizations are based on the color value and
% saturation of each pixel for black and white
% respectively, and the hue of the color for
% the other six color divisions.

% Originally coded at:
%http://www.owlnet.rice.edu/~elec301/Projects02/artSpy/mat
%lab.html#color
% Moodified by: Nizam Khan, 2005

function [hist_mat,pixels] = colorhistf(picname)

%Reads in and processes image
I = imread(picname);
I = im2double(I);

%Rips out colormap from image
[index,map] = rgb2ind(I);

%Computes total number of pixels in image
pixels = prod(size(index));

%Converts RGB values to HSV ones
hsv = rgb2hsv(map);


%Stores hue values
h = hsv(:,1);
```

```
%Stores saturation values
s = hsv(:,2);

%Stores color value values
v = hsv(:,3);

%Finds location of black and white pixels
darks = find(v < .2)';
lights = find(s < .05 & v > .85)';

%Sets flag in hue of white and dark pixels so they won't
be %counted for another
%color as well
h([darks lights]) = -1;

%Gets the number of all pixels for each color bin
black = length(darks)/pixels;
white = length(lights)/pixels;
red = length(find((h > .9167 | h <= .083) & h ~= -
1))/pixels;
yellow = length(find(h > .083 & h <= .25))/pixels;
green = length(find(h > .25 & h <= .4167))/pixels;
cyan = length(find(h > .4167 & h <= .5833))/pixels;
blue = length(find(h > .5833 & h <= .75))/pixels;
magenta = length(find(h > .75 & h <= .9167))/pixels;

%creating the histogram
hist_mat = zeros(8,1);
hist_mat(1,1) = black;
hist_mat(2,1) = white;
hist_mat(3,1) = red;
hist_mat(4,1) = yellow;
```

```
hist_mat(5,1) = green;
hist_mat(6,1) = cyan;
hist_mat(7,1) = blue;
hist_mat(8,1) = magenta;
```

%find_min.m

```
%function file that calculates least 5 distances
%Written by: Nizam Khan, 2005

function retimage = find_min(diffmat)

retimage = [];
for loop = 1:5
    ret_image_val = min(diffmat);
    ret_image_ind = find(diffmat == ret_image_val);
    retimage = [retimage ret_image_ind(1,1)];
    diffmat(ret_image_ind,1) = realmax;
end
```

```
% image_database.m
    %function file that creates the image database
    %Written by: Nizam Khan, 2005

    function [hist_mat,pix]= image_database()

    image_num = 31;
    hist_mat = zeros(8,image_num);
    pix = zeros(image_num,1);
    [hist_mat(:,2),pix(2,1)] = colorhistf('601.jpg');
    [hist_mat(:,3),pix(3,1)] = colorhistf('602.jpg');
    [hist_mat(:,4),pix(4,1)] = colorhistf('603.jpg');
    [hist_mat(:,5),pix(5,1)] = colorhistf('604.jpg');
    [hist_mat(:,6),pix(6,1)] = colorhistf('605.jpg');
    [hist_mat(:,7),pix(7,1)] = colorhistf('606.jpg');
    [hist_mat(:,8),pix(8,1)] = colorhistf('607.jpg');
    [hist_mat(:,9),pix(9,1)] = colorhistf('608.jpg');
    [hist_mat(:,10),pix(10,1)] = colorhistf('609.jpg');
    [hist_mat(:,11),pix(11,1)] = colorhistf('610.jpg');
    [hist_mat(:,12),pix(12,1)] = colorhistf('800.jpg');
    [hist_mat(:,13),pix(13,1)] = colorhistf('801.jpg');
    [hist_mat(:,14),pix(14,1)] = colorhistf('802.jpg');
    [hist_mat(:,15),pix(15,1)] = colorhistf('803.jpg');
    [hist_mat(:,16),pix(16,1)] = colorhistf('804.jpg');
    [hist_mat(:,17),pix(17,1)] = colorhistf('805.jpg');
    [hist_mat(:,18),pix(18,1)] = colorhistf('806.jpg');
    [hist_mat(:,19),pix(19,1)] = colorhistf('807.jpg');
    [hist_mat(:,20),pix(20,1)] = colorhistf('808.jpg');
    [hist_mat(:,21),pix(21,1)] = colorhistf('809.jpg');
    [hist_mat(:,22),pix(22,1)] = colorhistf('368.jpg');
    [hist_mat(:,23),pix(23,1)] = colorhistf('369.jpg');
    [hist_mat(:,24),pix(24,1)] = colorhistf('370.jpg');
```

```
[hist_mat(:,25),pix(25,1)] = colorhistf('371.jpg');
[hist_mat(:,26),pix(26,1)] = colorhistf('372.jpg');
[hist_mat(:,27),pix(27,1)] = colorhistf('373.jpg');
[hist_mat(:,28),pix(28,1)] = colorhistf('374.jpg');
[hist_mat(:,29),pix(29,1)] = colorhistf('375.jpg');
[hist_mat(:,30),pix(30,1)] = colorhistf('376.jpg');
[hist_mat(:,31),pix(31,1)] = colorhistf('377.jpg');
```

```
%   image_recog.m
%   Function file that implements histogram-based search
%   written by Nizam Khan, 2005
function image_recog(histmat,pixels)

    warning off;
    im_name = inputdlg('please enter the name of test
image:');
    image_name = char(im_name);
    [histmat(:,1),pixels(1,1)] = colorhistf(image_name);

    database_size = size(histmat);
    database_size = database_size(1,2);
    %histogram intersection with normalization
    diff_mat = zeros(database_size-1,1);
    for n = 1:database_size-1
        diff_mat(n,1) = sum(abs(histmat(:,1) -
histmat(:,n+1)));
    end
    %measuring similarity for histogram intersection
    ret_image = find_min(diff_mat);
    %printing similar image files
    image_show(ret_image);
    %
    temp_mat = zeros(8,database_size);
    for n = 1:database_size
        temp_mat(:,n) = histmat(:,n)*pixels(n,1);
    end
    % Calculating histogram euclidean method
    E_mat = zeros(database_size-1,1);
```

```
for n = 1:database_size-1
        E_mat(n,1) = sum((temp_mat(:,1) -
temp_mat(:,n+1)).^2);
end
    E_image = find_min(E_mat);
    figure(2)
    image_show(E_image);
```

```
% image_show.m
% function file to show similar images from the database
% written by Nizam Khan, 2005
function image_show(ret_im)

for loop = 1:5
    if (ret_im(1,loop) == 1)
        subplot(1,5,loop)
        imshow('601.jpg');
    elseif (ret_im(1,loop) == 2)
         subplot(1,5,loop)
         imshow('602.jpg');
    elseif (ret_im(1,loop) == 3)
         subplot(1,5,loop)
         imshow('603.jpg')
    elseif (ret_im(1,loop) == 4)
         subplot(1,5,loop)
         imshow('604.jpg')
    elseif (ret_im(1,loop) == 5)
         subplot(1,5,loop)
         imshow('605.jpg');
    elseif (ret_im(1,loop) == 6)
         subplot(1,5,loop)
         imshow('606.jpg');
    elseif (ret_im(1,loop) == 7)
         subplot(1,5,loop)
         imshow('607.jpg');
    elseif (ret_im(1,loop) == 8)
         subplot(1,5,loop)
         imshow('608.jpg');
    elseif (ret_im(1,loop) == 9)
         subplot(1,5,loop)
```

```
        imshow('608.jpg');
    elseif (ret_im(1,loop) == 10)
        subplot(1,5,loop)
        imshow('610.jpg');
    elseif (ret_im(1,loop) == 11)
        subplot(1,5,loop)
        imshow('800.jpg');
    elseif (ret_im(1,loop) == 12)
        subplot(1,5,loop)
        imshow('801.jpg');
    elseif (ret_im(1,loop) == 13)
        subplot(1,5,loop)
        imshow('802.jpg');
    elseif (ret_im(1,loop) == 14)
        subplot(1,5,loop)
        imshow('803.jpg');
    elseif (ret_im(1,loop) == 15)
        subplot(1,5,loop)
        imshow('804.jpg');
    elseif (ret_im(1,loop) == 16)
        subplot(1,5,loop)
        imshow('805.jpg');
    elseif (ret_im(1,loop) == 17)
        subplot(1,5,loop)
        imshow('806.jpg');
    elseif (ret_im(1,loop) == 18)
        subplot(1,5,loop)
        imshow('807.jpg');
    elseif (ret_im(1,loop) == 19)
        subplot(1,5,loop)
        imshow('808.jpg');
    elseif (ret_im(1,loop) == 20)
```

```
        subplot(1,5,loop)
        imshow('809.jpg');
    elseif (ret_im(1,loop) == 21)
        subplot(1,5,loop)
        imshow('368.jpg');
    elseif (ret_im(1,loop) == 22)
        subplot(1,5,loop)
        imshow('369.jpg');
    elseif (ret_im(1,loop) == 23)
        subplot(1,5,loop)
        imshow('370.jpg');
    elseif (ret_im(1,loop) == 24)
        subplot(1,5,loop)
        imshow('371.jpg');
    elseif (ret_im(1,loop) == 25)
        subplot(1,5,loop)
        imshow('372.jpg');
    elseif (ret_im(1,loop) == 26)
        subplot(1,5,loop)
        imshow('373.jpg');
    elseif (ret_im(1,loop) == 27)
        subplot(1,5,loop)
        imshow('374.jpg');
    elseif (ret_im(1,loop) == 28)
        subplot(1,5,loop)
        imshow('375.jpg');
    elseif (ret_im(1,loop) == 29)
        subplot(1,5,loop)
        imshow('376.jpg');
    elseif (ret_im(1,loop) == 30)
        subplot(1,5,loop)
        imshow('377.jpg');
```

```
      end
end
```

```
%proclpc.m
function [aCoeff,resid,pitch,G,parcor,stream] =
proclpc(data,sr,L,fr,fs,preemp)

% USAGE: [aCoeff,resid,pitch,G,parcor,stream] =
% proclpc(data,sr,L,fr,fs,preemp)

% This function computes the LPC (linear-predictive
% coding) coefficients that describe a speech signal.
% The LPC coefficients are a short-time measure of
% the speech signal which describe the signal as the
% output of an all-pole filter. This all-pole filter
% provides a good description %of the speech articulators;
% thus LPC analysis is often used in speech recognition
% and speech coding systems. The LPC parameters are
% recalculated, by default in this implementation, every
% 20ms.

% The results of LPC analysis are a new representation of
% the signal s(n) = G e(n) - sum from 1 to L a(i)s(n-i)
% where s(n) is the original data. a(i) and e(n) are the
% outputs of the LPC analysis with a(i) representing the
% LPC model. The e(n) term represents either the speech
% source's excitation, or the residual: the details of the
% signal that are not captured by the LPC coefficients.
% The G factor is a gain term.

% LPC analysis is performed on a monaural sound vector
% (data) which has been sampled at a sampling rate of "sr"
% The following %optional parameters modify the behaviour
% of this algorithm.
% L - The order of the analysis. There are L+1 LPC
```

```
% coefficients in the output array aCoeff for each frame
% of data. L defaults to 13.
% fr - Frame time increment, in ms. The LPC analysis is
% done starting every
% fr ms in time. Defaults to 20ms (50 LPC vectors a
% second)
% fs - Frame size in ms. The LPC analysis is done by
% windowing the speech
% data with a rectangular window that is fs ms long.
% Defaults to 30ms
% preemp - This variable is the epsilon in a digital one-
% zero filter which
% serves to preemphasize the speech signal and compensate
% for the 6dB
% per octave rolloff in the radiation function. Defaults
% to .9378.

% The output variables from this function are
% aCoeff - The LPC analysis results, a(i). One column of L
% numbers for each frame of data
% resid - The LPC residual, e(n). One column of sr*fs
% samples representing the excitation or residual of the
% LPC filter.
% pitch - A frame-by-frame estimate of the pitch of the
% signal, calculated by finding the peak in the residual's
% autocorrelation for each frame.
% G - The LPC gain for each frame.
% parcor - The parcor coefficients. The parcor
% coefficients give the ratio between adjacent sections in
% a tubular model of the %speech articulators. There are L
% parcor coefficients for each frame of speech.
% stream - The LPC analysis' residual or excitation signal
```

```
% as one long vector.
% Overlapping frames of the resid output combined into a
% new one-dimensional signal and post-filtered.

% The synlpc routine inverts this transform and returns
% the original speech signal.

% This code was graciously provided by:
% Delores Etter (University of Colorado, Boulder) and
% Professor Geoffrey Orsak (Southern Methodist University)
% It was first published in:
% Orsak, G.C. et al. "Collaborative SP education using the
% Internet and MATLAB" IEEE SIGNAL PROCESSING MAGAZINE
% Nov. 1995,vol.12, no.6, pp.23-32
% Modified by: Nizam Khan, 2005

if (nargin<3), L = 13; end
if (nargin<4), fr = 20; end
if (nargin<5), fs = 30; end
if (nargin<6), preemp = .9378; end

[row col] = size(data);
if col==1 data=data'; end

nframe = 0;
% Convert ms to samples
msfr = round(sr/1000*fr);
% Convert ms to samples
msfs = round(sr/1000*fs);
duration = length(data);
```

```
% Preemphasize speech
speech = filter([1 -preemp], 1, data)'; msoverlap = msfs -
msfr;
% Compute part of window
ramp = [0:1/(msoverlap-1):1]';

% frame rate=20ms
for frameIndex=1:msfr:duration-msfs+1

% frame size=30ms
frameData = speech(frameIndex:(frameIndex+msfs-1));
nframe = nframe+1;

% Compute the correlation
autoCor = xcorr(frameData);
autoCorVec = autoCor(msfs+[0:L]);

    % Levinson's method
    err(1) = autoCorVec(1);
    k(1) = 0;
    A = [];
    for index=1:L
        numerator = [1 A.']*autoCorVec(index+1:-1:2);
        denominator = -1*err(index);
        k(index) = numerator/denominator; % PARCOR coeffs
        A = [A+k(index)*flipud(A); k(index)];
        err(index+1) = (1-k(index)^2)*err(index);
    end

   aCoeff(:,nframe) = [1; A];
   parcor(:,nframe) = k';
```

```
    % Calculate the filter response by evaluating the
    % z-transform
    if 0
        gain=0;
        cft=0:(1/255):1;
        for index=1:L
            gain = gain + aCoeff(index,nframe)*exp(-
i*2*pi*cft).^index;
        end
        gain = abs(1./gain);
        spec(:,nframe) = 20*log10(gain(1:128))';
        plot(20*log10(gain));
        title(nframe);
        drawnow;
    end
    % Calculate the filter response from the filter's
    % impulse response (to check above).
    if 0
        impulseResponse = filter(1, aCoeff(:,nframe), [1
zeros(1,255)]);
        freqResp = 20*log10(abs(fft(impulseResponse)));
        plot(freqResp);
    end
     % find excitation noise
     errSig = filter([1 A'],1,frameData);
     % gain
     G(nframe) = sqrt(err(L+1));
     % calculate pitch & voicing %information
     autoCorErr = xcorr(errSig);
     [B,I] = sort(autoCorErr);
     num = length(I);
     if B(num-1) > .01*B(num)
```

```
        pitch(nframe) = abs(I(num) - I(num-1));
    else
        pitch(nframe) = 0;
    end

    % calculate additional info to improve the compressed
    % sound quality
    resid(:,nframe) = errSig/G(nframe);
  % add residual frames using a %trapezoidal window

if(frameIndex==1)
     stream = resid(1:msfr,nframe);
else
     stream = [stream;
     overlap+resid(1:msoverlap,nframe).*ramp;
     resid(msoverlap+1:msfr,nframe)];
end
    if(frameIndex+msfr+msfs-1 > duration)
        stream = [stream; resid(msfr+1:msfs,nframe)];
    else
        overlap = resid(msfr+1:msfs,nframe).*flipud(ramp);
    end
end
stream = filter(1, [1 -preemp], stream)';
```

%updateCenter.m

```
% ====== Find new centers (This is the same as the one in
% kmeans.m.)
% Originally written by: Roger Jang, Sept 24, 1996.
% found at:
%http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/DCPR/
% Modified by: Nizam Khan, 2005

function [center, distortion, distMat, U] =
updateCenter(center, dataSet, distMat)

centerNum = size(center, 1);
dataNum = size(dataSet, 1);
dim = size(dataSet, 2);

% ====== Find the U (partition matrix)
[a,b] = min(distMat);
index = b+centerNum*(0:dataNum-1);
U = zeros(size(distMat));
U(index) = ones(size(index));

% ====== Check if there is an empty group (and delete
% them)
index=find(sum(U,2)==0);
emptyGroupNum=length(index);
if emptyGroupNum~=0,
     fprintf('Found %d empty group(s)!', emptyGroupNum);
     U(index,:)=[];
end
```

```
% ====== Find the new centers
center = (U*dataSet)./(sum(U,2)*ones(1,dim));
distMat = vecdist(center, dataSet);

% ====== Add new centers for the deleted group
if emptyGroupNum~=0
     distortionByGroup=sum(((distMat.^2).*U)');
     % Find the %indices of the centers to be split
     [junk, index]=max(distortionByGroup);
     index=index(1:emptyGroupNum);
     temp=center; temp(index, :)=[];
     center=[temp; center(index,:)-eps;
center(index,:)+eps];
     distMat = vecdist(center, dataSet);
     [a,b] = min(distMat);
     index = b+centerNum*(0:dataNum-1);
     U = zeros(size(distMat));
     U(index) = ones(size(index));
     center = (U*dataSet)./(sum(U,2)*ones(1,dim));
     distMat = vecdist(center, dataSet);
end

% objective function
distortion = sum(sum((distMat.^2).*U));
```

```
%vecdist.m
function distmat = vecdist(mat1, mat2)

% VECDIST Distance between two set of vectors
% VECDIST(MAT1, MAT2) returns the distance matrix between
% two set of vectors MAT1 and MAT2. The element at row i
% and column j of the return matrix is the Euclidean
% distance between row i of MAT1 and row j of MAT2.

% Originally written by: Roger Jang, Sept 24, 1996.
% Found at
%http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/DCPR/
% Modified by: Nizam Khan, 2005

if nargin == 1,
     mat2 = mat1;
end

[m1, n1] = size(mat1);
[m2, n2] = size(mat2);

if n1 ~= n2,
     error('Matrices mismatch!');
end

distmat = zeros(m1, m2);

if n1 == 1,
     distmat = abs(mat1*ones(1,m2)-ones(m1,1)*mat2');
elseif m2 >= m1,
     for i = 1:m1,
```

```
          distmat(i,:) = sqrt(sum(((ones(m2,1)*mat1(i,:)-
mat2)').^2));
     end
else
     for i = 1:m2,
          distmat(:,i) = sqrt(sum(((mat1-
ones(m1,1)*mat2(i,:))').^2))';
     end
end
```

%vqLBG_bb.m

```
function [center, U] = vqLBG_bb(dataSet, codeBookSize,
plotOpt)

% vqLBG: Vector quantization using LBG method of center
% splitting
% Usage: [center, U]=vq(dataSet, codeBookSize, plotOpt)
% dataSet : sampleNum x featureDim.
% codeBookSize : codebook size or number of cluster
% centers (should be the power of 2)

% Originally written by: Roger Jang
% found at:
%http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/DCPR/
% Modified by: Nizam Khan, 2005

if nargin==0, selfdemo; return; end;
if nargin<3, plotOpt=0; end;

% ====== Error checking
[sampleNum,featureDim] = size(dataSet);
if sampleNum<=1, msgbox('Error on dataSet!'), return; end
if codeBookSize<1, msgbox('Error on codebook size!'),
return; end
if codeBookSize>sampleNum/2, error('codeBookSize must be
less than half of the row size of dataSet!'), end

% Initial Centers: mean of all data
center= mean(dataSet);

if plotOpt
     plot(dataSet(:,1), dataSet(:,2), 'b.');
```

```
     centerH=line(center(:,1), center(:,2), 'color', 'r',
'marker', 'o', 'linestyle', 'none', 'linewidth', 2);
     axis image
end;

% Do while-loop to increase center number till it is equal
% to or greater than codebook size.
maxLoopCount = 100;
centerNum = size(center,1);
while (centerNum<codeBookSize)
     if plotOpt
          fprintf('Hit return to start center
splitting...\n'); pause;
     end
     center=[center-eps; center+eps];
     centerNum = size(center,1);
     prevDistortion = realmax;
     distMat=vecdist(center, dataSet);
     for i = 1:maxLoopCount
          [center, distortion, distMat, U] =
updateCenter(center, dataSet, distMat);
          if(abs((prevDistortion-
distortion)/prevDistortion) < eps)
               break;
          else
               prevDistortion = distortion;
          end
          if plotOpt
               set(centerH, 'xdata', center(:,1), 'ydata',
center(:,2));
               drawnow;
          end
```

```
     end
end

if plotOpt
     color = {'r', 'g', 'c', 'y', 'm', 'b', 'k'};
     figure;
     plot(dataSet(:, 1), dataSet(:, 2), 'o');
     maxU = max(U);
     clusterNum = size(center,1);
     for i=1:clusterNum,
          index = find(U(i, :) == maxU);
          colorIndex = rem(i, length(color))+1;
          line(dataSet(index, 1), dataSet(index, 2),
'linestyle', 'none', 'marker', '*', 'color',
color{colorIndex});
          line(center(:,1), center(:,2), 'color', 'r',
'marker', 'o', 'linestyle', 'none', 'linewidth', 2);
     end
     axis image;
end
```