MODELING AUDIO DATA FOR MULTI-CRITERIA QUERY FORMULATION

BY

TIZETA ZEWIDE

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

JULY, 2007

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

MODELING OF AUDIO DATA FOR MULTI-CRITERIA QUERY FORMULATION

BY

TIZETA ZEWIDE

Name and Signature of members of the Examining Board:

1. Dr. Solomon Atnafu, Advisor _____
2. _____ _____
3. _____ _____

ACKNOWLEDGMENT

I want to thank Graduate Committee of the Department of Computer science for giving me the permission to commence this research work.

This thesis owes its existence to the help, support, and inspiration of many people. It is a pleasant aspect that I have, now, the opportunity to express my gratitude for all of them.

In the first place, I would like to express my sincere appreciation to my advisor Dr. Solomon Atnafu for his guidance, support and advise throughout the course of this thesis's work.

I acknowledge my gratitude to physicians at Black Lion Hospital for their collaboration and valuable assistance in the research.

I am grateful to Dr. Hailubeza Alemu who kept any eye on the progress of my work and always available whenever I needed his advices and support. My sincere thanks go to Abera Abebaw for editing my thesis, providing me invaluable insights and eliminating many shortcomings on the implementation issues. I would also like to thank Getnet Abebe for reviewing my thesis and offering me indispensable feedbacks.

Finally, I owe special gratitude to my family for their continuous and unconditional love, help and encouragement throughout my study.

Table of contents

List of Figures

List of Tables

Abstract

The amount of available audio data in multimedia databases is increasing rapidly in consequence of advancements in media creation, storage and compression technologies. This rapid increase imposes new demands in audio data management and retrieval. As a result, growing number of research works have been put into the area, such as audio indexing, classification and segmentation. However, modeling and retrieval techniques are not adequate and handling audio data content is still far from sufficient for most retrieval tasks.

This work proposes an audio data and audio repository model to fulfill user requirements in retrieving audio data from large collections. The audio data repository model we proposed in this thesis enable us to capture the audio itself, its low-level representation, all alphanumeric data associated to the audio as well as timing information. Thus, it facilitates both keyword-based and similarity-based operations on audio objects. In the proposed model, a generic audio repository model that can handle a general audio as well as a sub-repository model (speech repository model) that can manipulate speech through its constituent units is discussed. The speech repository model enable us to capture all relevant information associated to speech units, to keep track of hierarchical relationships between speech units and the speech that contains them. The Object Relational scheme is used to manage these audio related data under a DBMS.

The proposed work augments audio retrieval by enabling users to apply semantic concepts (high-level features) linked to audio signals in their query in order to match relevant audio in a database. Such an approach improves simple matching based on audio signal characteristics (low-level features), as the user need not have example sounds for querying. In addition, an example audio can be used as a query since users may not always know exact search terms to achieve useful results.

Finally, the practicality of the proposed model is demonstrated by taking sample application areas from the medical domain.

Keywords: low-level features, high-level features, audio data model, audio repository model, content-based audio retrieval, keyword-based audio retrieval, Query-By-Example (QBE), ADMMA.

1. Introduction

As computational resources are becoming less of a bottleneck to digitally record and store vast amounts of multimedia data, the amount of archived materials in digital form is also increasing rapidly-and this amount continues to grow. Yet, the value of this information depends on how easily we can manage, find and access it. As a result, content extraction, indexing and retrieval of multimedia data continue to be one of the most challenging and fastest-growing research areas.

Many of the multimedia documents that are available today in profusion on the Internet or in private archives contain an audio part. Audio data is an integral part of many modern computers and multimedia application as a result it has become a critical component of information systems that needs to be efficiently managed. However, for years it has often been overlooked and little attention has been directed towards audio while multimedia research efforts predominately focused on image and video analysis.

This thesis addresses some of the requisite audio modeling and retrieval issues and introduces an effort aimed at integrating such techniques in practical systems. It proposes audio models that capture and structure the most significant properties of audio so as to enable multi-criteria query formation and retrieval. Multi-criteria query formation, in this context, means that users are allowed to query audio collections either using example audio data or keywords. In the case of example audio queries, audio signal is represented by a more compact abstract numerical representation that is referred to as low-level feature of the signal. On the other hand, in the case of keyword-based retrieval, domain dependent keywords that characterize semantic representation of the signal are considered. Finally, a prototype system incorporating the proposed models has been built for retrieving audio information in the medical domain.

1.1. Motivation

Audio, as an independent media, carries rich information from which one can derive semantic understanding. For instance, while a text summary of a meeting can describe the decisions reached during a meeting, using it together with sound bites conveys several information (e.g. the emotional tone at the time those decisions were made). Nowadays, various applications areas such as: Bioacoustics, Music industries, Investigation services (surveillance, human

characteristics recognition, forensics, intruder detection), and many more similar ones require an interaction with large audio collection. For instance, the identity of a speaking person can be identified from a database of audio recordings. In Bioacoustics, one can identify animals based on the sounds they produce. In the case of automatic surveillance, approaching cars that cannot be seen by mounted cameras can be detected.

Audio signals are integral parts of many modern computer and multimedia applications. They enclose information that can be used to index and retrieve the documents they belong to (e.g. whistles and audience expressions can be good indicators of play break in sport videos). Audio components can also be seen as a fundamental aspect of audiovisual productions. Large digital audio collections of sound effects are used by the movie and animation industry and around 75% of them are added at post-production [1]. Most of the movies produced today use large amount of sounds taken from large libraries of sound effects. Whenever we hear a door closing, a gunshot or a telephone ringing in a movie it almost always originates from one of those libraries [2]. These indicative but by no means exhaustive lists of application areas show the need for tools to manipulate, analyze and retrieve audio signals from large digital audio collections.

Stimulated by this ever-growing availability and need of audio materials to the user via new media and ways of distribution (e.g. Internet) an increasing need to identify and classify audio data has emerged. As an indication of the magnitude of such a collection, the number of CD tracks in retail is more than 3 million [2]. However, given the enormous amount of available audio material it has become more and more difficult for the consumer to locate audio data that fits his or her personal tastes. In order to understand the need for modeling and retrieval of audio for enabling multi-criteria query formation and retrieval, let us see the following scenarios.

Application Scenario

In this section, the gist of the proposed work will be presented by describing some possible usage scenarios in medical applications. These scenarios are indicative of the necessities of interacting with audio collections and show how the system we are going to describe meets the requirements of the application domain. Here, the emphasis is on how the system can be used rather than how

it works. One might want to return to this section after reading the subsequent chapters that explain how the proposed system works to see again how they fit in the application context.

Heart sound retrieval

Assume that a patient who has a heart related problem arrives at a hospital. The physician tries to examine the patient's heart sound so as to determine what problems he/she has. At some point, the physician hears a strange sound while listening through a stethoscope. The physician is able to understand what he hears is an abnormal heart sound but doesn't know what it is. Therefore, the physician decides to record the patient's heart sound he/she heard and submit to an audio retrieval system. For instance, imagine a system that, given an input of a heart sound, could return the label "Murmur", and given an input prompt of "Normal heart sound" could retrieve from a database samples most like that of a normal heart sound. The physician checks if there are similar sounds at the reference sounds stored in the database and see what related issues can be considered.

Image description

As a continuation from the previous scenario, the physician decides to send the patient to a cardiologist to get a brief description regarding his/her heart situation. The cardiologist, on the receiving end, makes echocardiography, records his description of echo findings and sends the recording together with the image to the requesting physician. The benefit of having image descriptions in an audio format is that communication through audio is natural, easy, fast and can convey more information than using mere text. As a result, it is an ideal technique to be used to describe the overall information of the image in a short time and a more descriptive way. The requesting physician, after having the recording, wants to know if there are problems at the "Mitral valve" of the heart. He is not interested to hear the entire recording rather he is only interested in hearing descriptions related to the "Mitral valve". So, he selects words that are more representative to the problem he suspects and is able to retrieve only segments of the audio recording that goes in line with his/her interest.

1.2. Problem statement

Audio data, with its unique characteristics such as huge size, rich content, and temporal nature, has posed many interesting challenges to the multimedia research community. Although a large amount and variety of audio signals have been available on the internet and stored in different repository disks, information retrieval methods for those signals are still in their infancy. Audio retrieval has been an important and challenging research field for more than fifteen years. Although the research community yielded great technical advances in the past, work in this area is still at a preliminary stage and there are still unsolved problems to deal with. The main problem in audio researches emanate from the inherent characteristics of audio, which is linearity. For example, it is difficult to locate the discussion of a particular topic in an audio recording of a two hour meeting since it requires the user to search linearly or sequentially through the recorded audio. The other problem is that, the representation of audio signals is currently based on numerical features at a low-level abstraction that does not consider semantic information. In relation to the representation of audio data, representation of queries in retrieval systems is also an additional problem that worth mentioning. Last but not least, modeling of audio data for enabling user access capabilities such as in querying and retrieval that incorporates the semantics of audio together with its low-level representations is one of the problems that are overlooked in audio researches. While data modeling is the basis for any multimedia information management system, to the author's best knowledge, no research attempt has been made towards modeling audio data.

Various researches raised issues towards content-based audio retrieval. Early approaches employed query-by-example techniques. In these research communities, audio-based queries such as query-by-humming have gained importance particularly in the field of music retrieval [3,4,5]. Such systems lack semantic interpretation as retrieval is merely based on acoustic similarity. Other attempts in audio retrieval focus on metadata, which is data about data. In this case, retrieval is based on manual index terms or keywords, a method familiar to users used to web search engines [6]. The major drawback of such systems is that they are subjective, expensive and insufficient to characterize the media.

1.3. Objectives

General objective:

To develop a data model and repository model for capturing audio data content and enabling multi-criteria query formation and retrieval.

Specific objectives:

To meet the aforementioned general objective, the following specific objectives are set.

- Identify the components that need to be captured in order to represent and describe an audio data.
- Design a generic audio data model.
- Design an audio repository model
- Identify appropriate low-level audio features that can be applied for the application domain considered.
- Develop a prototype to demonstrate that the proposed model can be used for adequate audio data management and multi-criteria query formation and retrieval.

1.4. Scope of the study

The scope of this study is limited to proposing a data model, repository model and demonstrating the practicability of the model to be used in a multi-criteria query formation and retrieval. Other areas such as audio classification and topic segmentation will not be dealt with in this study though their inputs and contribution are enclosed in the proposed models. Audio in the course of this study doesn't consider mixed sound signals and refers to a single signal (either speech, music or environmental sounds).

1.5. Thesis organization

The remainder of this thesis is organized as follows: In Chapter 2, the background information regarding audio analysis and a survey of related works are presented. Chapter 3 addresses the proposed models. Experiments and results are described in Chapter 4. In Chapter 5, an Audio Data Management for Medical Application (ADMMA) prototype that demonstrates the

applicability of our proposal is presented. In Chapter 6, conclusions and future works are depicted.

2. Literature review

This work falls under the umbrella of computer audition, which is the study of algorithms, techniques and system design for the purpose of extracting useful information from audio data. In light of this, the sections to follow present general concepts that are believed to give generic insight of audio related issues and works that are directly related to this thesis work.

2.1. Background

2.1.1. Digital Audio

Audio, in the context of this work, is defined as a signal whose frequency range is within the human audible range (approximately 20 Hz to 20 KHz) [77]. Audio can be captured using devices such as microphone or produced by program algorithms. Since physical sound is analog, it has to be digitized in order to be processed with digital hardware. Audio recording devices take analog or continuous signal, such as the sound picked up by a microphone or sound recorded on magnetic media, and convert it into digital values with specific audio characteristics such as format, encoding type, number of channels, sampling rate, sample size, compression type etc. In order to enable a perfect reconstruction of the digital signal, the analog signal has to be sampled uniformly and at a frequency that is equivalent to at least twice its bandwidth [7]. The analog signal is sampled at uniform intervals and quantized into a digital code. Quantization always introduces some noise, known as quantization noise that is not necessarily audible. A widely known example for digitally encoded analog audio is the CD-Audio standard. It defines a sampling rate of 44.1 KHz and a quantization of 16 Bits. Such an encoding preserves all perceivable frequencies and does not introduce audible quantization noise.

2.1.2. Information Retrieval (IR)

In times gone by, IR research has been concerned with searching documents in a database by textual query and is familiar to many through the popular web search engines such as Google. The classic IR problem is to locate desired text documents using search query consisting of a number of keywords. Typically, matching documents are found by locating query keywords

within them. If a document has high number of query terms, it is regarded as being more "relevant" to the query than other presented to the user for further exploration, as the web search engines do.

In the last decades, the number of available multimedia archives has grown. Though powerful IR algorithms are available for text, traditional text-based information retrieval might not be appropriate for retrieval of audio and multimedia in general. Information retrieval from audio data is sharply different from information retrieval from text. Unlike text documents, audio poses a new challenge due to its sequential nature: it is laborious to scan through multiple long stories to obtain specific information of direct relevance to a given query. One solution may be to rely on annotated meta-data information attached with audio. Manual annotation of multimedia objects by humans is not always applicable as it is time-consuming, error-prone and costly. Furthermore, this information is often incomplete or not available at all. These limitations of metadata based retrieval techniques can be overcome by examining the content of media objects as well. Content-based information retrieval is a separate branch of research of information retrieval where information about multimedia documents is extracted directly from their content. In such cases, there is no need for a priori knowledge concerning the documents. However, information retrieval based on the content of a media lacks consideration of semantic information that is conveyed by the media object.

2.1.3. Audio representation

Audio signals are data-intensive signals and are stored (in their basic uncompressed form) as series of numbers corresponding to the amplitude of the signal over time. Basically, there are two fundamental methods used in representing audio data: the time and the frequency domain representation. The time domain (time-amplitude) representation (Figure 2-1) shows the amplitude of the signal as a function of time. It is the sampling of the audio signal that comes into the audio sensing device. The frequency domain (frequency-magnitude) shown in Figure 2-2, on the other hand, shows the frequency components and frequency distribution of a sound signal. The frequency-magnitude can be derived by computing the Fourier transform[1] of the signal in the time domain. Although time-domain representation is adequate for transmission and reproduction

[1] Fourier transform is a reversible integral transform of one function into another.

of arbitrary waveforms, alone it is not appropriate for analyzing and understanding audio signals as it fails to show when the different frequency components occur. In the same manner, frequency domain representation lacks showing the variation of amplitude over time. As a result, sometimes, a combined representation or a form of time-frequency analysis technique called a spectrogram is used. The spectrogram of a signal basically represents the energy distribution of the signal in a time-frequency plane. In this representation, time and frequency components are shown in the same representation. Frequency content is shown along the vertical axis, time along the horizontal one and the intensity (power) is indicated by the gray scale – darkest part indicating greatest amplitude/power (Figure 2-3).

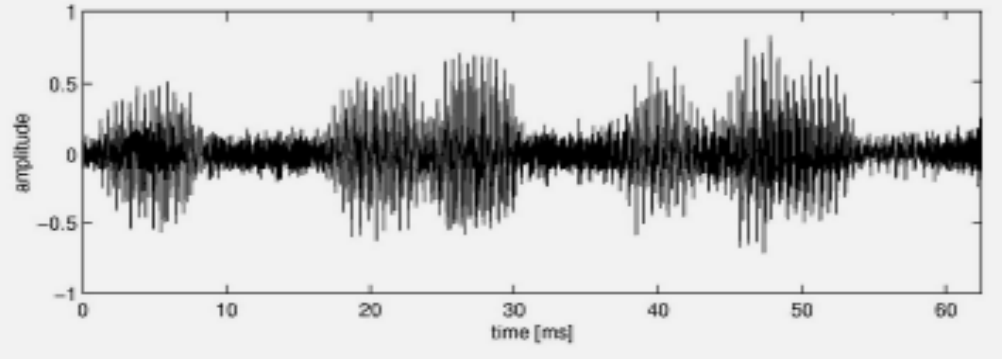


Figure 2-1: Audio signal: time domain [35]

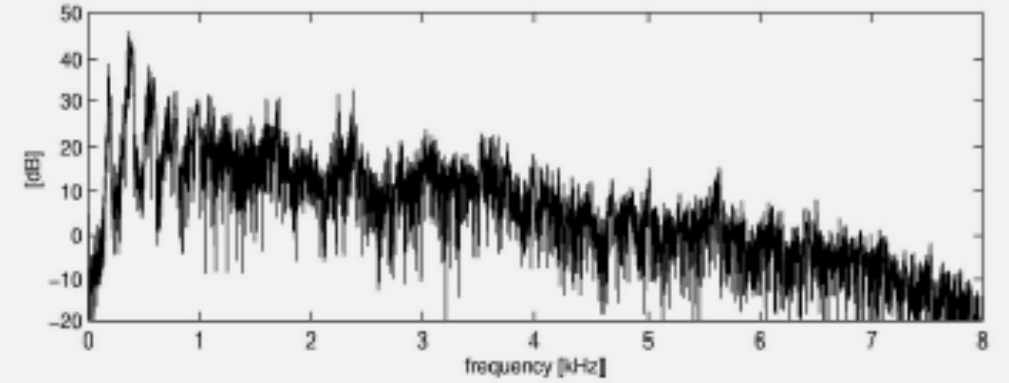


Figure 2-2: Audio signal: frequency domain [35]

One good example of a time-frequency analyzer is a human ear as the early stages of Human Auditory System (HAS) decomposes incoming sound wave in the cochlea into different frequency bands [2].

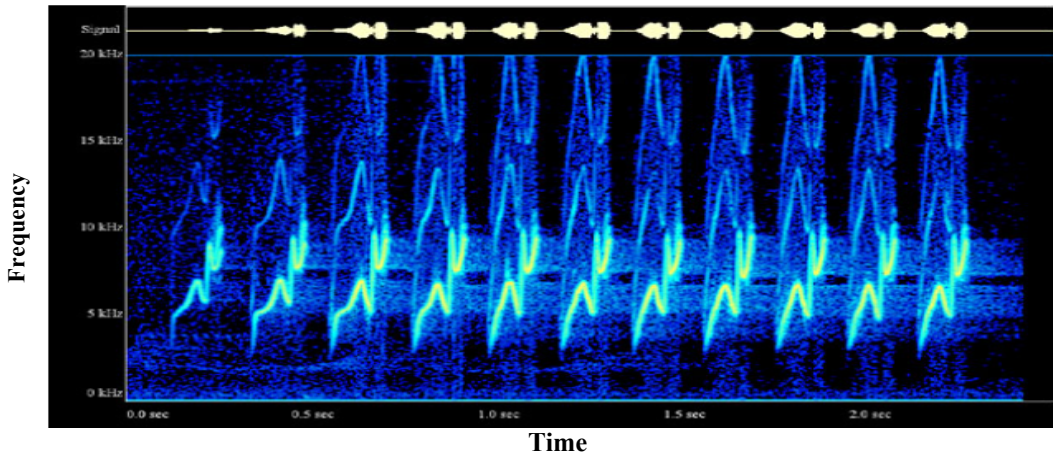


Figure 2-3: Time-Frequency analysis (Spectogram) [35]

2.1.4. Audio Features

Audio features are those characteristics that describe the contents of an audio signal. These features enable us to reduce the amount of data to be processed as well as variability of audio signal in time [8]. Most of the time, audio features are numerical values that are derived from a segment of audio. The process of computing those values is known as feature extraction. This numerical representation, which is called the feature vector, has a fixed dimension and therefore can be thought of as a point in a multidimensional feature space. The feature vector is subsequently used as a fundamental building block of various types of audio analysis and information extraction algorithms as it is used to characterize, classify and index a given audio signal [2].

It is typical for audio analysis algorithms to be based on features computed on frame basis (i.e. the signal is broken in small chunks called analysis windows) (Figure 2-4). Their sizes are usually around 20 to 40 milliseconds [2]. For such short duration of the window, the signal is assumed to be stationary and its characteristics are relatively stable. Many features have been proposed in the literature. Some of them are designed for specific tasks, while others are more general and can be useful for a variety of applications. Most of the time, evaluation of audio features is done in the context of some specific application or analysis technique. For example, a set of features for representing speech might be evaluated in the context of speech analysis while a set of features for representing musical content might be evaluated in the context of automatic musical genre classification [2].

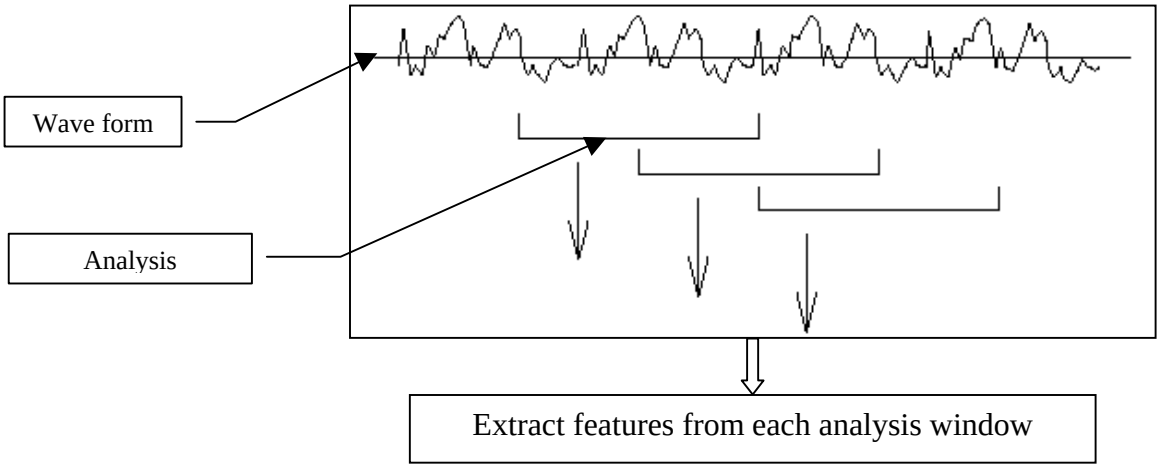


Figure 2-4: Audio Analysis Window

The features used in computer audition are divided into several categories. In fact, there is no widely accepted taxonomy of audio features. A basic approach is to consider the domain of the feature. Time-based features are extracted from the signal in time domain. Spectral features are derived after the signal has been transformed using one of the basic signal processing transforms such as Fourier, Cosine, and Wavelet transforms [9]. In the following section, we review some of these features that can be directly extracted from a given audio representation (time-domain, frequency-domain or time-frequency domain) and used in most of audio analysis tasks. The goal is to identify suitable features that can be used for generic audio data.

Time domain audio features

- **Power** is the most straightforward physical feature that measures the amplitude of a waveform at any one time. The louder the signal the more the power is in it. The distribution of power across time has been used to distinguish between speech and music. Speech signals have variable power distribution while music tends to have more consistent distribution.
- **Average energy** is a measure for the loudness of the signal.
- **Zero Crossing Rate (ZCR)** is a signal feature that measures how often a waveform crosses zero per unit time. It can be used to discriminate between speech and music as their ZCR contour[2] varies significantly [10]. The most attractive feature of ZCR is that it

[2] Change of ZCR over time

can be calculated in real time without even having analog to digital conversion as it needs only identifying sign changes of voltage in a waveform.

- **Rhythm** is a perceptual quantity that indicates perceivable events in the sound that repeat in a predictable manner. It relates to the rate, regularity and pattern of time-level events like drum beats, note and linguistic events like prosodic emphasis and rhyme. It can be used to detect rhythmic sound effects such as footsteps, clock ticks, telephone rings and the likes [11].

Frequency domain features

- **Bandwidth** is defined as the measure of the range of frequencies present in the waveform. It is used to discriminate between speech and music [10, 12] as music has a larger bandwidth than speech.

- **Fundamental frequency (f_0)** is the lowest frequency at which the signal repeats itself. It is relevant only for periodic or pseudo-periodic signals. f_0 is an important feature for distinguishing between pieces of music or for retrieving pieces of music based on melody. In speech, f_0 is used to indicate word boundaries [13].
- **Pitch** is one of the most important perceptual features that is closely related to the physical feature f_0. While frequency is an absolute numerical quantity, pitch is a relative quantity. For instance, a musical pitch is understood in a relative scale (i.e. the relationship between notes is much important than their absolute location on a frequency scale).
- **Harmonicity** refers to the relationship between peaks in the spectrum. It is mainly used to differentiate between voiced and unvoiced speech.
- **Timbre** measures the quality of a sound produced by a particular voice or instrument (e.g. it is what makes a violin sound like a violin, which is different from a saxophone). The main problem is timbre is a subjective measure and it is rather difficult to quantify.
- **Frequency Centroid (FC)** is also a commonly used audio feature that represents the center of gravity of the spectrogram. Its effectiveness in characterizing the spectral information is demonstrated in [14].

- **Mel Frequency Cepstral Coefficients (MFCC),** sometimes categorized as a time-frequency feature, is a well-established acoustic feature derived from the energy spectrum that is capable of capturing varied spectral phenomenon. Human beings do not perceive all frequencies in the same way. Therefore, a transformation is often used such that the contribution of different frequencies is weighted in a way it corresponds better to the human hearing. The resulting coefficients for a given signal are called the MFCC. Though intended to model speech MFCCs have been applied successfully in non-speech tasks such as the music system developed in [15] and more general audio studies in [16,17]. This success indicates that they are good starting points for general audio discrimination.

- ***MPEG-7 Low-Level Descriptors (LLD)***

MPEG-7 provides concepts that describe multimedia data. The audio part of the MPEG-7 standard specifically contains descriptive elements that characterize the underlying audio signal itself rather than merely "labeling" it with high-level tags (as is frequently associated with the name "metadata") [19]. MPEG-7 Audio provides structures for describing audio content (e.g., spectral, parametric, and temporal features of a signal), and high-level Description tools that are more specific to a set of applications (More will be said about MPEG-7 in section 2.1.6). MPEG-7 low-level audio descriptors are of general importance in describing audio. There are seventeen of them that can be used in a variety of applications. They can be roughly divided into the following groups: Basic, Basic Spectral, Timbral Temporal, Timbral Spectral, Spectral Basis as well as Silence Descriptor (Figure 2-5). These descriptors can be extracted from audio automatically and depict the variation of properties of audio over time or frequency. Based on these descriptors it is often feasible to analyze the similarity between different audio files [16].

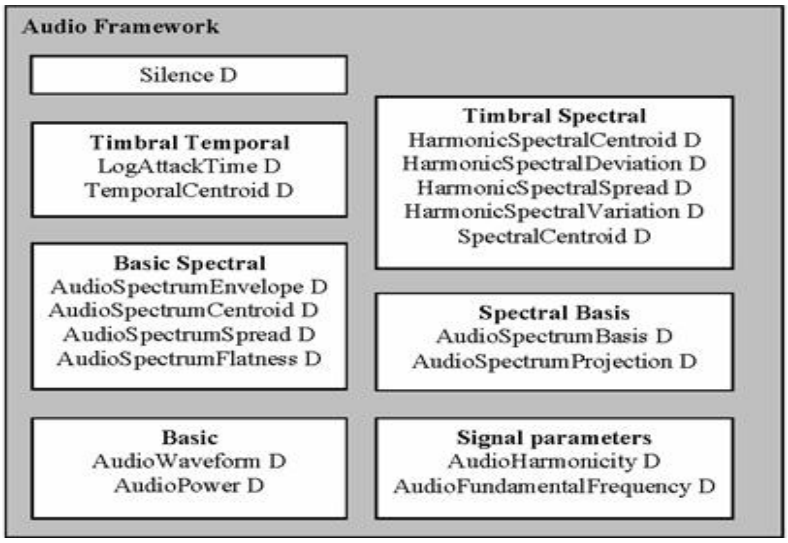


Figure 2-5: MPEG-7 LLDs [76]

In the following sections, we describe the MPEG-7 Audio LLDs together with their perceptual meaning. A more detailed description can be found in [20].

Basic

The LLDs in the *Basic* group primarily enable a short description of the shape of an audio waveform. The *AudioWaveform* (AW) descriptor represents the waveform envelope and is mainly intended for economic display of a waveform in an audio editor. AW comprises the minimum and maximum values of a framed signal. The *AudioPower* descriptor computes the average square of the waveform samples in a frame. It describes the power of the signal over time [19,20].

Basic Spectral

The LLDs in the *BasicSpectral* group describe basic properties of the spectrum of an audio signal. The *AudioSpectrumEnvelope* (ASE) descriptor represents the short-term power spectrum of a signal with a logarithmic frequency scale in several frequency bands. The logarithmic frequency scale aims at imitating properties of the human ear. The ASE descriptor is the basis for the computation of the other descriptors in the *BasicSpectral* group. The *AudioSpectrumCentroid* (ASC) is the center of gravity of the spectrum calculated by ASE. The ASC descriptor indicates whether high or low frequencies dominate the spectrum of the signal. The *AudioSpectrumSpread*

(ASS) descriptor represents the deviation of the power spectrum from its centroid. ASS enables separation of tonal sounds from noise-like sounds. The fourth descriptor of the *BasicSpectral* group is *AudioSpectrumFlatness* (ASF). ASF describes the deviation of the spectrum of an audio signal from a flat shape. A flat spectrum indicates a noise-like or impulse-like signal. According to the MPEG-7 standard ASF is designed to perform *fingerprinting*, which requires robust matching between pairs of audio signals [19,20].

Spectral Basis

The *SpectralBasis* descriptors *AudioSpectrumBasis* (ASB) and *AudioSpectrumProjection* (ASP) are techniques for general-purpose sound recognition. ASB transforms the spectrum of a signal to a much lower-dimensional representation under certain statistical constraints. The ASB descriptor is based on the power spectrum, similarly to the ASE descriptor and provides a compact representation of a spectrum, while preserving a maximum amount of information. The ASP is used together with the ASB descriptor. ASP takes a decibel-scaled spectrum as input and projects it against spectral basis functions, previously computed by ASB [19,20].

Signal Parameters

The *SignalParameters* group contains the *AudioFundamentalFrequency* (AFF) descriptor and the *AudioHarmonicity* (AH) descriptor. The AFF descriptor represents the fundamental frequency of a sound. AFF can be applicable to sound segmentation of speech and music. AH is a measure for the degree of harmonicity in a signal. The descriptor comprises of two components: *harmonic ratio* and *upper limit of harmonicity*. The harmonic ratio is the proportion of harmonic components in a signal. A purely harmonic signal has a harmonic ratio of “1”, while the harmonic ratio of noise is “0”. The upper limit of harmonicity specifies the frequency beyond which the audio signal has no more significant harmonic components [19,20].

Timbral Temporal

Timbral descriptors are usually employed in Music Information Retrieval (MIR). Timbre is a sound property that is independent of pitch and loudness. The *LogAttackTime* (LAT) characterizes the attack of a sound. The attack time is the time from the beginning of a sound

signal to a point in time where its amplitude reaches a maximum. LAT is the logarithm of the attack time. The attack characterizes the beginning of a sound, which can be smooth or sudden. LAT can be employed for classification of musical instruments. The *TemporalCentroid* (TC) is the point in time where most of the signal energy is located [19, 20].

Timbral Spectral

Harmonic peaks in a spectrum correspond to frequencies that are a multiple of the fundamental frequency. They are appropriate to describe the timbre of a signal. The *TimbralSpectral* descriptors rely on harmonic peak estimation by the fundamental frequency of the audio signal. The *HarmonicSpectralCentroid* (HSC) is the amplitude-weighted average of the harmonic peaks in a spectrum. The *HarmonicSpectralSpread* (HSS) descriptor is the amplitude-weighted deviation of the harmonic peaks from the HSC. The *HarmonicSpectralDeviation* (HSD) is the deviation of the harmonic peaks from the spectral envelope. The spectral envelope is the mean over a few neighboring harmonic peaks. *HarmonicSpectralVariation* (HSV) refers to the correlation of harmonic peaks in adjacent frames. The fifth *TimbralSpectral* descriptor is *SpectralCentroid* (SC), which is the power-weighted average of the frequencies in the power spectrum. The timbre descriptors are usually applied to music information retrieval in which timbre plays an important role [19, 20].

Feature selection

An audio archive might contain a broad range of sound types which are distinguished by different acoustic characteristics. In such cases, the major difficulty to be faced is extracting features that are capable of characterizing all sounds in those archives. One solution is to identify those features that are applicable to the majority of sounds classes as it will be unrealistic to make use of all audio features in the literature. Broad selections of acoustic features have been applied with varying success on different tasks. Besides, considerable studies have been carried out in order to find optimal features that are applicable to the general audio and specific audio class discriminations [17, 21]. But these studies showed that the optimal features selection depends on the domain and classification technique. As the feature compositions are optimized for a specific

domain, they are unlikely to scale well to more complex discrimination tasks and general audio analysis.

In order to facilitate good characterization, an acoustic feature should ideally distinguish between significant acoustic variations and yet eliminate irrelevant spectral detail and noise which do not contribute to recognition. As discussed in the previous section, MFCCs are well established acoustic features that can suppress undesirable spectral variation and be applied in general audio studies particularly when the number of audio classes is large [22]. On the other hand, various studies have shown that MPEG-7 low-level descriptors are comparable to their MFCC counter parts. Moreover, MPEG -7 low-level descriptors are tailored to be used in generic audio signals. Therefore, to avoid redundancy, only MPEG -7 features are considered in this study.

2.1.5. Audio classification

Audio signal classification is concerned with identifying to which set of classes a sound is most likely to fit. Classification of audio signals has different steps (Figure 2-6). The first step of the classification process consists of extracting a set of audio features from the signal. The extracted set of audio features is then compared with a set of previously stored statistical models of the audio classes, one for each of the candidate classes. These models can be obtained by analyzing a high number of representative audio examples from each class, which were subjected to the same feature extraction process as the unknown input examples. The features of these training examples were being reduced to a simple statistical description by using modeling techniques such as Gaussian Mixture Model (GMM), Hidden Markov Model (HMM) etc. The class of an unknown input audio example is, then, declared as the one with the statistical model that best describes its set of extracted features.

The basic drive behind classification is to classify a given stream of audio into possible categories because:

- The various audio classes require different processing, indexing and retrieval techniques
- The various classes have different significance to different applications
- Search space will be reduced to a particular audio class during the retrieval process

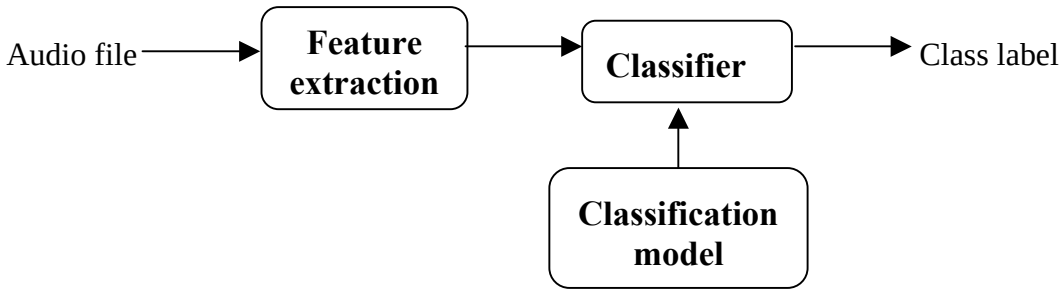


Figure 2-6: Audio classification steps

In recent years, a substantial literature on audio classification has been developed. Approaches mainly differ in the set of acoustic features used to represent the audio signal and the classification technique applied. Most current researches only involve limited classes of sounds e.g. discrimination between music and speech or classification among silence, music, speech and noise etc.[21, 23]. Several techniques have been employed for the purpose of classifying an unknown sound. The principle is to measure similarity between an input feature vector and those of known sounds. In the early days of speech processing, template matching between feature vectors was used intuitively. Current acoustic research favors stochastic models, which provide more flexibility and more theoretically meaningful likelihood scores. Of these, the most common approaches are *(GMM)* based methods [16, 21, 17], *(HMM)* [11], *Nearest Neighbor* methods [21, 17], *Neural Network (NN)* variants [15], *Vector Quantization (VQ)* [15, 16] and S*upport Vector Machine (SVM)* [23, 24].

Different researchers have used various categories of audio in audio content analysis. In [18] audio data is coarsely divided in to three classes, namely, music, speech and environmental sounds. Music analysis mainly deals with the identification, classification and retrieval of music genre, artist, instruments and structures [25]. Speech recognition and analysis has also a long tradition and is a matured research area that focuses in the identification and recognition of input speech signals. The third category is the class of environmental sounds. For further discussion on classification of general sound, refer to [26].

A more general classification system can differentiate between speech, music and other environmental sounds. Then, based on the result of classification, different processing, indexing or retrieval techniques will be applied accordingly. For instance, if a sound is classified as

speech, speech recognizer will be employed, and if it is music, music transcriber will be called into service.

2.1.6. MPEG-standards

The MPEG audio coding standard is an example of a perception-based coder that exploits the characteristics of the human auditory system in order to compress audio more efficiently. MPEG audio compression is a lossy perceptual-based compression scheme that allows audio files to be stored in approximately one tenth of the space they would normally require if they were uncompressed [27]. This enables large numbers of audio files to be stored and streamed over networks. One favorable factor in using MPEG compression technique is that it can be used to compress any type of audio. In the first step the audio signal is converted to spectral components. Each spectral component is, then, quantized and coded with the goal of keeping the quantization noise below the masking threshold. Simultaneous masking is a frequency domain phenomenon where low-level signal (the maskee) can be made inaudible (masked) by a simultaneously occurring stronger signal (the masker) as long as masker and maskee are close enough to each other in frequency [2,28,29].

Due to the ever-increasing amount of multimedia material, which is available to users, efficient management of such material by means of content-related techniques is of growing importance. This goal can be achieved by using pre-computed descriptive data ("metadata"), which is associated with the content. One example of such metadata standards for audiovisual data is the MPEG-7, which is one component of the MPEG standard. MPEG-7 aims to define a standard for describing multimedia documents, which is to represent information about the content of the document and not the content itself unlike MPEG-1, MPEG-2, and MPEG-4 [30,31].

MPEG-7, formally known as Multimedia Content Description Interface, specifies a standard set of descriptors that can be used to describe various types of multimedia information. It defines a wide framework for the description of audio visual and generic properties of multimedia content, covering both high-level semantic concepts as well as low-level features. An interesting feature of MPEG-7 is that it is not devoted to a particular application; the description provided is able to support a range of applications as broad as possible. The basic descriptive entities in MPEG-7 that define the syntax and the semantics of the feature representation are called *Descriptors*(D).

Description Schemes (DS) are intended to combine the pre-defined structures of Descriptors and their relationships. Both Descriptors and Description Schemes are syntactically defined by a so-called *Description Definition Language* (DDL), which also provides the ability for future extension of existing elements. The MPEG-7 DDL is based on XML Schema as the language of choice for the textual representation of content description and for allowing extensibility of description tools [31, 32, 33].

2.2. Related works

In this section, the different works that are related to the premise of this thesis are presented. The works that have been given emphasis are data modeling and audio retrieval issues as they are directly related to this study.

2.2.1. Data models

Data modeling is the process of structuring and organizing data. It defines a structure for data that is typically implemented in a database management system. It can also be used to limit data entry to data that fits in the specified structure [34]. The richness of the data model plays a key role in the usability of information retrieval system as errors in data models may lead to irrelevant or incorrect answers to user queries. The role of a data model in a Database Management System (DBMS) is to provide a framework to express the properties of the data items that are to be stored and retrieved using the system. In Multimedia Database Management Systems (MMDBMS), the data model assumes an additional role of specifying and computing different levels of abstraction from multimedia data. Moreover, a multimedia data model must deal with the issue of representing the multimedia objects, which is designing the high and low-level abstractions of the media data to facilitate various operations. These operations may include media objects indexing, browsing, querying, retrieval and communication. A good data model should fulfill the following issues [35]:

- A representation should be provided for the logical media structure. It is essential to represent this structure explicitly both for querying and for representation.
- Semantic meaning should be modeled and linked to low-level characteristics and the structure of the media.

- The meta-data necessary for the operation of the system components needs to be determined and stored in the database.
- It is necessary to rely on international standard, e.g., MPEG-7, in order to guarantee interoperability for data sharing and data exchange [35].

2.2.1.1. Audio data model

Audio is becoming an essential part of the information systems and multimedia applications. As large amounts of audio data have become publicly available, the need to model and query these data efficiently happens to be significant. An audio data model is a set of concepts that can be used to describe the structure and content of an audio. This model should be extensible and have the expressive power to present the structure and contents of the audio signal. Audio data models are different from traditional data modeling methods in that they add additional features that stem from the unstructured and continuous properties that are inherent to them. Consequently, the audio data model is one of the main issues in the design and development of any audio database management system.

The process of audio description consists of extracting audio characteristics, recognizing the audio objects and assigning semantics to these objects. To the author's knowledge, no audio data model has been proposed in other research works. As a result, only data models regarding image and video data model are revised in this section.

2.2.1.2. Image data models

Different works have been proposed to represent the content and semantic richness of an image. Visual Information Management System (VIMSYS)[36], Adaptive Image Retrieval (AIR)[37] and a generic image model that is proposed in [38] are some of the valuable image data models proposed by previous researches.

VIMSYS model views the image information entities in four planes. This model is based on the image characteristics and the inter-relations between those characteristics in an object-oriented design. These planes are the Domain Objects and relations (DO), the Domain Events and relations (DE), the Image Objects and relations (IO) and the image representations and relations (IR). Objects in this model have a set of attributes and methods associated with them. They are

connected in a class attribute hierarchy. The attribute relationships are spatial, functional and semantic. The **IO plane** has three basic classes of objects: images, image features and feature organizations. These objects are related to one another through set-of, generalization (is-a), and relations. Image feature is further classified into texture, color, intensity and feature-of geometric feature. The **DO plane** consists of a semantic levels specification of domain entities, built upon the two previous levels. The **DE plane** has been included in the model to accommodate the event definition over image sequences. The **IR plane** is clearly functional. The top three layers form the content of the image. Though this model has addressed the need to use image objects in retrieval, it has given less emphasis on the semantic and context representation of images.

AIR model claims that it is the first comprehensive and generic data model for a class of image application areas that coherently integrates logical image representations. It is a semantic data model that facilitates the modeling of an image and the image-objects in the image. It can be divided into three layers: physical level, logical level and semantic or external level representation. There are two kinds of transformations that occur in the model. The first is the transformation from the physical to the logical representation, such as a spatial oriented graph. The second transformation involves the derivation of the semantic attributes from the physical representation [37].

A generic image data model proposed in [38] presents an image data model that describes an image in several levels of abstraction considering both the content of the image and the semantic information associated to it unlike the aforementioned models. It also considers the various types of relations like the temporal, spatial and semantic relations between the salient objects within the image. This model basically comprises the external and content space. The external feature captures the general information that are assumed to be external to the content of the image (context, domain and image oriented information). The content subspace, on the other hand, represents the image in terms of the physical, spatial and semantic features together with the relations among salient objects of the image.

2.2.1.3. Video data models

Various researches have been made in the area of video modeling/retrieval based on audiovisual content, such as color, texture and motion [39, 40, 41]. The advantage of this approach is that

features can be extracted automatically. The low-level schemes, however, are very limited in expressing high-level information. In contrast, video data models based on semantic content [42, 43, 44] are capable of supporting more "natural" queries. They, however, must rely partially on manual annotation. A limitation of this approach is that semantic content can be ambiguous and context dependent. A video data model that considers both the visual content and semantic information is proposed by [45]. The proposed video model is a generic one, which exhibit the structural decomposition (scene-shot-frame) of a video clip as well as the object based representation (Scene-AVO[3]-VOP[4]) to entertain video in raw bitstreams and in MPEG-4 compressed bitstreams respectively. In this model, in addition to the content of the video, the metadata is also modeled, classified and represented.

2.2.2. Audio retrieval

Audio Information Retrieval (AIR) is the process of finding audio information in audio databases by making use of available resources. If the available resources are series of keywords annotated manually then the retrieval is said to be keyword-based retrieval. On the other hand, if the available resource is a piece of audio, the retrieval is known as content-based retrieval. Different reasons can be attributed to the need of AIR: availability of audio information in the form of music, news reporting, comedy audio segments, online radio and sports broadcasts are some to list [46]. However, identifying an audio segment that satisfies user's requirement and human perception has been an outstanding issue for the past two and three decades. Human beings have amazing ability to distinguish different types of audio. Given any audio piece, one can instantly tell the type of audio (e.g., human voice, music, or noise), speed (fast or slow), mood (happy, sad, relaxing, etc), and determine its similarity to another piece of audio. Nonetheless, a computer sees a piece of audio simply as a sequence of sample values.

A lot of different approaches have addressed automatic analysis of audio content, be it speech/music classification [21, 47, 48], retrieval of similar sounds [49, 12], or music genre classification [15]. The main topic of this paper, however, is to present a system which performs identification and retrieval of audio signals rather than assigning them to predefined categories. As described above, two approaches have been commonly employed to retrieve audio data. The

[3] Audio Visual Objects
[4] Video Object Planes

first is to use content-based retrieval, where the query is a piece of audio by using a similarity measure. The other approach is to generate textual indices automatically, semi-automatically or manually and then use the traditional information retrieval. The subsequent subsections describe research works made in the area of audio retrieval using low-level (primitive) and high-level (semantic) features.

2.2.2.1. Audio retrieval based on primitive features

Primitive features or low-level features are called so because most of them are extracted directly from digital representations of objects and have little or nothing to do with human perception. These features are most suitable for applications that use the audio signal directly as in the case of investigation services. In the past, researches have extensively addressed primitive features in multimedia database models and algorithms to search for relevant data. Such retrieval approaches are known as Content-based retrieval. Content-based retrieval tools offer different functionalities. One functionality is the query-by-example (QBE) approach, where the main method of specifying a query is by example. In this paradigm, a user can provide an example sound (e.g. a piece of music) or utter as a query to the system without having contents annotated and labeled, possibly restricting the search to a non-semantic subspace. QBE is usually done in the following ways: first, features are extracted from the example. Second, the distances between the feature vectors of the example and the database samples are estimated using a certain distance metric. Finally, database samples having the shortest distance to the example are retrieved.

A classic example of QBE is the query by humming method used to retrieve music by humming a melody which is used in [5]. The retrieval algorithm browses a database composed of recorded musical pieces, and sorts them according to their similarity with the hummed melody. In this paper, the authors' approach hinges upon the observation that Melodic Contour(MC)[5] can be used to discriminate between melodies as it is believed that MC is one of the most important methods that listeners use to determine similarities between melodies. An alphabet of three possible relationships between pitches (`U', `D', and `S'), representing the situation that a note is above (U), below (D) or the same(S) as the previous note. They used autocorrelation to isolate and track the peak energy level of the signal which is a measure of the pitch. Thus a user input audio is first converted into a sequence of relative pitch transitions. Songs in the database are also

[5] The pattern of ascending and descending pitch changes in a melody.

preprocessed to convert the melody into a stream of U,D,S characters, and the converted user input query(the *key*) is compared with all the songs using pattern matching techniques. While this technique is effective for retrieval of music scores such queries are not particularly natural or convenient for other sound types.

A query-by-example system for isolated sounds has been developed at Muscle Fish LLC [12]. The Muscle Fish system (later developed into the commercial application SoundFisher) is a pioneering work that has resulted in a compelling audio retrieval for a general database by similarity demonstration. Its approach is to analyze sound files for a specific set of acoustic features that include loudness, pitch, bandwidth and harmonicity. In this regard, this work is completely different from the previous one in that it has made use of low-level features that are considered to be representative to the audio signal. These features, then, result in a vector of attributes of those features. The system, then, measures similarity between a new sound and sounds in a database and ranks them based on their proximity. As a result, users of the system can search for and retrieve sounds by acoustical features, can specify classes based on these features and can ask the engine to retrieve similar or dissimilar sounds. In [50] an audio retrieval system is developed for detecting specific passages within a piece of music on any other audio file. This system is also based on a query-by-example paradigm, where the user selects a reference passage and asks the system to retrieve perceptually similar occurrences.

In recent years, the topic of “audio fingerprinting”, which can be considered as part of QBE, has been receiving more and more scientific and industrial interest. Audio fingerprint can be seen as a short summary of an audio object that can be used to establish the perceptual equality of two audio pieces. First, fingerprint is extracted from a set of known audio material and stored in a fingerprint database. Unknown content can then be identified by comparing the signature to the signatures contained in the database [51, 52, 53]. In order to investigate the potential behind fingerprint robustness and scalability, the authors of [51] have used short audio excerpts from songs having different musical genres by subjecting them to different signal distortions like time scale modification, mp3 encoding, amplitude compression etc. keeping the “original” as a baseline. Similarity is measured by comparing the number of bit errors between the derived fingerprints against a certain threshold.

The advantage of the QBE approach is that similarity is computed on features that are automatically derived from the audio signal (annotations are not required) and can therefore be applied inexpensively on a large scale. But these systems make no pretence of attaching semantics to the queries as they are not oriented towards audio semantics. Moreover, it is often hard to initialize the first query because the user may not have a good example to begin with. In addition to the aforementioned problems, even in the presence of an example sound, most of the audio data retrieving challenges arise from low signal quality due to far-field microphones, as found in courtrooms, and effects of other sound sources, background music, crowd noise , room acoustics and the likes[54]. For instance, the problem with regard to multiple overlapping speakers, as found in meetings, is common in the speech analysis community. Due to these reasons most researches on speech are particularly based on idealized data - read or laboratory speech. Such speech is `impoverished' with respect to phenomena that occur in the speech that is used every day. The difference between spontaneous and laboratory speech is that the former contains significant rates of disfluencies (e.g. pauses, repetitions and repairs) [55]. These limitations force audio retrieval mechanisms not to rely only on the low-level features rather to incorporate the high-level semantics of audio signals in their design.

2.2.2.2. Audio retrieval based on Semantic (high-level) features

Semantic features are powerful enough to describe multimedia content in general and audio in particular at varying levels of complexity and can support robust semantic expression that highlights wide range aspects of multimedia content. Audio information retrieval based on semantic features is certainly a reasonable answer to the semantic drawbacks of information retrieval based on primitive features. Having got used to text retrieval engines such as Google, users may prefer to query the database by keyword. Many systems with keyword annotations can provide such kind of services [6]. Recent works in audio retrieval have shown an interesting shift from QBE to query by keywords (QBK). So, it is apparent that modeling the high-level meaning of sound requires semantic content together with their low-level counterparts.

A very popular means of semantic audio retrieval is to annotate the audio with text, and use text-based database management systems to perform the retrieval. Annotation can be obtained in different ways. One method is to use manual annotation by taking text notes while the audio is being recorded along with the times when they were written. However, this approach has

significant drawbacks when confronted with large volumes of media data. Firstly, manual annotation is a time consuming task and might not always give the correct starting and ending time of a conversation because they do not always provide complete segments of the media for presentation. Secondly, it introduces subjectivity and may result in different semantic content as it needs human intervention. The other most commonly used approach is to apply Speech Recognition (SR) techniques to extract spoken words from a sound track. Manual annotation is still widely applicable because SR systems are mostly error prone due to problems such as Out Of Vocabulary (OOV) words, challenges related to independent speakers and continuous speech, lack of language models and poor audio quality etc [56]. In addition, SR systems' application is limited to speech.

Regardless of manual or automatic annotation, the common difficulty is the problem that emanates from natural languages. Natural languages mostly induce imprecision and ambiguity in that they present polysemy – "bike" can mean both "bicycle" and "motorcycle" – and synonymy– both "elevator" and "lift" refer to the same concept. This obliges users to guess how the given audio is annotated, which results in either too many or too few results to be returned [57]. Last but not least, annotation based retrieval might not be appropriate due to the inherent features of some audio signals. For instance, information regarding background events such as laughter, music and the likes are lost during transcription process which results in incompleteness of the audio data. Therefore, for applications that demand precise details of the original audio, transcription methods are inappropriate.

In response to the aforesaid problems, one alternative is identifying domain dependent keywords that mostly represent a given audio and associate those keywords with the audio. Keyword- based annotation approaches are mostly superior to full-annotation based systems in cases where domain dependent applications are considered. As indicated previously, one major problem of SR techniques is OOV words. Terms that are common ("technical") to specific application areas might not be included in the vocabulary of generic SR systems. Even if it is possible to develop domain specific SR systems, keyword-based systems can also be used in their absence (doesn't imply replacement). Thus, the keyword-based audio retrieval system proposed in this work can be used in unconstrained domain provided that domain experts are involved in the keyword selection.

Various research works used text transcription that can be derived automatically from an audio segment. Such an approach has been investigated in the NIST Spoken Document Retrieval (SDR) evaluation [58,59]. The authors applied text-based retrieval methods to documents that are produced by a large vocabulary speech recognition system. A research work presented in [60] used a method that bridges the semantic gaps between low-level features and high-level semantics to facilitate semantic indexing and retrieval. The authors proposed a hierarchical approach that models the statistical behaviors of audio events[6]. Four representative audio events: gunshot, explosion, engine, and car-breaking are considered to represent and detect gun play and car-chasing scenes that are considered to express the high-level semantics. The authors also proposed two stages of models, i.e. audio event modeling and semantic context modeling, which are constructed to hierarchically characterize audio clips. A state-of-the-art audio retrieval system developed by Malcolm Slaney in 2002 [6,61] also demonstrates labeling and retrieval of audio samples.

2.3. Summary

As indicated above, most of the approaches mainly employ low-level audio features to model and index audio data, which may cause semantically unrelated data to be close only because they may be similar in terms of their low-level features. Furthermore, systems using only low-level features cannot be interpreted as high-level human perceptions. For example, given a query of a high-pitched bird song, a system based on acoustics might retrieve other high-pitched, harmonic sounds such as a door bell ringing.

On the other hand, annotation based systems have the same problem as conventional text information retrieval and due to their subjective nature of descriptions, it will be difficult to satisfy users query. Therefore, as audio is information rich, a single approach is not enough to capture all the contents of audio. One can observe that there is evidence of a growing synergy between traditional text-based and content-based retrieval techniques and notice that the development of systems that combine the two may yield better results. Thus, the essential property of the introduced retrieval system lies in the fact that it does not only rely on the low-level features of an audio data but also on the availability of metadata information that is associated to the audio signal. In order to make such systems practical, the audio data must be

[6] Short audio clips that represent the sound of an object

modeled to capture both the low-level and high-level audio information. The following chapter is dedicated to describe the proposed audio model.

3. Audio data model

In this chapter, we deal with the modeling aspect of audio data. As mentioned in section 2.2.1, a good model is essential in enabling a wider range of applications. Modeling audio data is the process of designing an abstraction of raw audio to facilitate various information retrieval and manipulation operations. It determines mainly what features are to be used in the storage and retrieval where other components, such as content analysis tools and query processing techniques, are also more or less dependent on it.

3.1. Proposed Audio data model

In the past, most of the research works have focused in modeling image and video data. As indicated in the previous sections, a lot of work has been done in the area of audio retrieval. But, to the author's best knowledge, no audio data model has ever been designed. However, in practical situation, the retrieval issue should have come next to audio modeling. This actuality indicates that audio data modeling is overlooked and any effort in such area is indisputable.

In the course of this study, we have used the audio classification presented in [18]. The authors of [18] classified an audio stream into speech, music and environmental sounds. Environmental sounds are assumed to include all audio signals that are considered to be non-speech and non-music. Each of these classes can be broken down further into more detailed descriptions even if there is no well defined taxonomical structure. The basic problem in structuring audio emanates from the heterogeneous characteristics of each group of audio classes. When speech is concerned, it is possible to hierarchically structure it in to topics, sentences and words. But the same classification can not be applied to other categories. For instance, structuring and classification of music is somewhat vague. How detailed can one categorize music? Limitations have to be done in order to get a realistic structure scheme that can be implemented. Simple structuring of music into a few set of classes is often used. Though it is difficult to structure music in a hierarchical fashion, it can be structured into its main components: intro, verse, chorus, bridge et cetera and each of these can be sub classified in to further classes

The aforesaid difficulties signify that as different audio types require different processing and retrieval techniques, a general audio signal is first classified into one of those audio classes. After

an audio signal is identified into one of the classes, descriptive audio features tailored to that specific audio group will be used based on the MPEG-7 standard. The basic determinants for classification are the audio features considered and the classification model used [18]. Audio classification, as discussed in section 2.1.5, is a matured research field. Using those classification techniques, general audio can be segmented and classified in terms of its semantic classes. General audio, in this context, is defined to be any audio clip with no assumptions on length, segmentation, source category or composition with other sounds (e.g. mixtures of speech and music).

This study doesn't take into account the problem of general audio organization by machines in order to support applications in audio classification and clustering. One can use any of the classification models in the literature. For instance, a number of tools exist within the MPEG-7 framework for representing category concepts. With these standardized schemes in hand, it is possible to share pre-trained probability models between applications for discrimination of audio classes. In this thesis, we rather focus in identifying and proposing an audio data and repository model that can be used for generic audio signals. Throughout this thesis work, we use MPEG-7 features for representing the high-level and low-level information because of the previously discussed advantages MPEG-7 offers. Several issues were considered in the due course of coming up with a convincing model, for instance:

- We have tried to represent and structure audio data in such a way that it becomes suitable environment for storage and query operations.
- Semantic meaning of the audio data is modeled together with the low-level representation that characterizes it.
- Metadata that are believed to be necessary for the operation of the audio model and retrieval system are determined in advance and stored in the database next to the media
- The proposed model is based on MPEG-7 standard in order to provide interoperability among systems and application in the management and consumption of audio data.

The proposed audio data model can be used as a generic model that can be applied in unconstrained domain. If particular consideration is needed to a specific audio class, further refinement is necessary. This classification is made based on the signal characteristics which are

represented by MPEG-7 low-level descriptors. These features are modeled together with their metadata information. In this work, we focus on structuring and modeling of speech and class of environmental sounds. Speech is structured into its constituent units: topics, sentences and words while environmental sounds are taken as unstructured indivisible audio classes.

The concept of metadata was introduced as an attempt to alleviate problems associated with manipulating large quantities of data. Metadata are designed to summarize data in a format that is compact so as to minimize storage costs, and optimize manipulations such as search and retrieval. Such descriptive information about audio data, which is delivered together with the actual content, would be one way to facilitate audio data search immensely. Metadata that describe multimedia are usually embedded within the media using a proprietary format. This makes them intricate to be searchable. So, in order to make them retrievable, there must be a consistent representation after extraction. This is where the standard multimedia descriptor interface MPEG-7 description plays its role. Representing the metadata information using MPEG-7 alleviates the problem of inconsistency and helps to have a common way of representation. In the work regarding musical knowledge management, which is presented in [62] musical metadata is classified into three categories depending on the nature of the source from where the information can be extracted. In our study, these metadata classifications are used with little modification. The metadata classifications used in the proposed audio data model are defined below.

Creation and Production Information Metadata (CPI metadata) – are metadata elements that are related to the process of audio data creation and application-specific information, such as creators (authors, singers), title, date recorded etc, which are usually used in the audio context. This information is usually provided by the media acquisition process. In this work, it is referred to as *CPI metadata*.

Contextual metadata – is subjective information which is an external knowledge produced by the environment (listener) resulting from patterns, categories or associations with previously known facts. These descriptions cannot be derived directly from the audio data, as interpretation is necessary. This information constitutes the overall impression that is created on the listener while he/she is listening to that specific audio. So, this contextual metadata enables to support audio data with content description, as it is a key in efficient semantics handling.

Storage metadata - refers to information directly related to the medium holding the audio and the characteristics of the audio encoding. This information is not produced by external elements but rather provides information on the storage characteristics and its related practical constraints. The audio data can have specific characteristics depending upon how the audio data was digitally recorded. Storage metadata includes information such as format, encoding type, compression type, number of channels, sampling rate, sample size etc. It contains a lot of useful information in that it enables users, for instance, to choose music with high or low sound quality from online music store depending on the speed of their Internet connection.

In general, by integrating the abovementioned categories of metadata, it is possible to cope with audio data efficiently. Along with those descriptions, one can capture low-level descriptors to make required piece of audio identified. Since MPEG-7 offers a framework for the description of audio documents that describe the actions, objects and context of an audio clip, all these types of information can be handled by MPEG-7 description schemes. Obviously, some of these descriptions like creation, production and usage can only be inserted manually whereas other content descriptions can partly be extracted automatically. Basic information about the storage media, like encoding, number of channels, sampling rate, sample size, can directly be read from the analog to digital converter media devices. The proposed audio data model that incorporates all the aforementioned issues is shown in Figure 3.1.

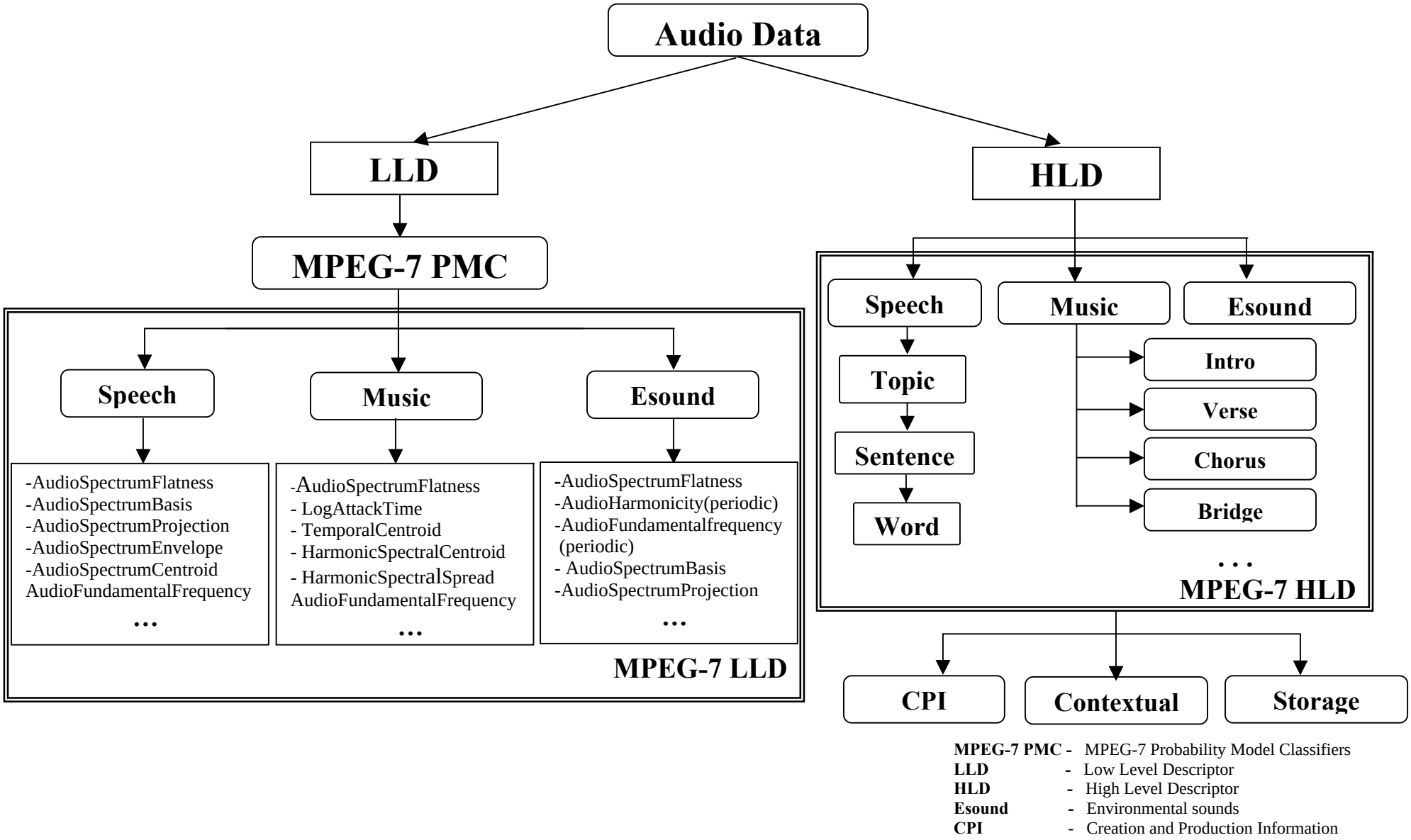


Figure 3-1: The proposed audio data model

In the proposed generic model, an audio data will first be classified in to one of the identified audio classes using MPEG-7 Probability Model Classifiers (PMC). Once a given audio is classified in to one of those categories, appropriate and specific analysis will be applied. To demonstrate our proposed model, we have considered two of the classes namely speech and environmental sound for further analysis. The major reason is the scope of the study (i.e. since each of these audio classes is a broad research area, considering all is unattainable). The other reason is, the application domain selected for this study. Among the different audio applications in the medical domain, the two audio classes under consideration are viewed to be practical as per the interest of domain experts in the area. In the medical applications, on the other hand, speech can be used as a means of communication among physicians. However, because of the difficulty of locating information in large audio archives, speech has not been valued as an archival source. In this study, we will demonstrate the applicability of speech in conveying medical information.

With regard to speech analysis, one approach is to apply Automatic Speech Recognition (ASR) techniques. However, as it is mentioned in section 2.2.2.2, ASR systems have limitations in addressing large vocabulary, speaker independent and continuous speech [56]. In recognition of such limitations, domain dependent keyword searching on the metadata information attached to speech is deemed to be a good choice. However, taking the whole speech as an indivisible unit and analyzing in its entirety will lead to flawed ending. To explain this, let us see a simple IR method. In IR techniques, frequency is a good indicator to decide whether a document is relevant to a query. The same is true for speech retrieval after it gets transcribed. But, consider a long speech document containing only one section relevant to a query. If a keyword is used only in the pertinent section, its overall frequency in the document will be low and, as a result, the document as a whole may be judged irrelevant despite the relevance of one section. The second reason is, once a section of speech is identified to be relevant, users would benefit from direct access to those sections. For example, if a user wants to find a particular news item in a database of radio or television news programs, he/she may not have the time and patience to endure a 30 minute broadcast to find the one minute clip that interests him/her. Therefore, dividing documents into sections based on their hierarchical structure addresses each of these problems.

In addition, it is possible to further segment a topic into sentences and sentences into words so as to use words as atomic units for speech retrieval. In this work, sentences are derived from a given

speech by identifying sentence and word boundaries through silence detection techniques. Silence detection, in turn, is done by measuring the energy levels of the speech signal against a given threshold. For example, a high threshold can be used to detect long pauses between sentences as silence while tolerating shorter breaks between phrases. A lower threshold can be used to identify phrasal pauses or even pauses between words. In this way, audio material can be split up into parts that reflect its physical structure, for instance speech into sentences and sentences into words. In our proposed model, once words are found then the sentences where they reside in can be obtained by tracking the temporal (beginning and ending time) information.

In modeling speech, one important issue that worth mentioning is the representation of low-level features. As low-level audio features are representations that are fully dependent on audio signals, it is difficult to rely on them in content-based speech retrieval. The reason is that differences in vocal tract lengths of individual speakers, which in turn result in differences among the various audio signal properties. As a result, low-level features extracted from different speakers might be different even if they are uttering the same word. Thus, we have focused only on the high-level features (semantics) which are to be captured from a given speech signal so as to show the applicability of the proposed model.

The class of environmental sound constitutes a number of audio signals that are neither speech nor music. In order to model them, apart from the low-level descriptors, domain-dependent keywords must first be extracted by domain experts. Finally, all the metadata information associated with the above mentioned audio units, together with their low-level features, are stored as MPEG -7 descriptions next to the media (speech) in the database. Audio processing is, then, applied on these descriptions (high-level or low-level) instead of the actual media.

3.2. Audio Data Repository Model (ADRM)

A data repository model is a conceptual representation of a repository that deals with the way data is stored in a Database Management System (DBMS). In the previous section, audio components that need to be captured and modeled are identified. This section describes how to represent audio components in a convenient way so as to store them in a DBMS.

The recent explosion in the use of media-rich applications has resulted in an appreciation for the value of multimedia content, and a realization of the challenges in managing that content. Relational data model has been very effective and efficient in storing and managing alphanumeric data. It has had enormous success ever since it has been designed. One of its popularity comes from its distinguished mathematical foundation that enables its easy theoretical treatment. In addition, this model is widely used because it is relatively easy to learn and use with high expressive power [35]. However, managing multimedia content presents unique issues and challenges on the existing DBMSs and this shortfall called for a paradigm shift to Object Relational DBMS (ORDBMS).

Based on the fact that most existing DBMSs are basically not designed for multimedia, database vendors provide extenders that enable fundamental processing of multimedia data (e.g., Oracle interMedia [63] and IBM InformixDataBlades [64]). These DBMSs have been extended to ORDBMS and allow integrated tools that can be used to manage multimedia data in addition to the traditional alphanumeric data management. Multimedia Database Systems (MMDBMS) such as ORDBMS organize and store multimedia data for content-based retrieval. Such systems rely on multimedia data models representing high-level and low-level abstractions of media objects for facilitating various operations like insertion, indexing, querying or retrieval. This powerful and well-established mechanism in the object-oriented world includes integral support for multimedia objects to provide the basis for adding complex objects, such as digitized audio, image, and video into databases. Due to this reason, in this thesis, we limited ourselves within the ORDB framework.

A table that holds audio data as an attribute is quite different from a relational table that store alphanumeric data. In relational model, items held in database tables are essentially abstract concepts or attributes, which describe external real world objects. On the other hand, media objects such as audio files are themselves real world objects that need to be held in or referred from database tables. Hence, audio data needs additional attributes that describe its peculiar characteristics through the use of keywords or low-level features. The database system that stores audio information should also have a way to store, manipulate and render audio in a way different from techniques employed in the relational systems. Besides, an audio data repository must capture the low-level feature descriptions and metadata information of audio data that are

required for content-based and high-level feature based (e.g. Keyword-based) audio retrieval. The following list shows some of the additional requirements to be considered in an audio data repository model.

- Audio is a content rich medium. Thus, its repository model must be able to capture both the low-level and high-level descriptions which are considered to give supplementary information. The low-level descriptors need to have compact representation so as to be used in content-based similarity matching.
- The different audio attributes call for new data structures to be defined.
- An audio repository model should be able to capture the temporal dimension of audio so as to take advantage of timing information.

Due to the aforementioned reasons, any contribution in designing and proposing an appropriate repository model for audio data is highly needed. Hence, in this thesis, we propose an audio data repository model that facilitates capturing, representation and management of audio data. Our proposed audio data repository model can also be considered as an implementation of the audio data model proposed in the previous section under an ORDBMS paradigm.

The proposed Audio Data Repository Model - Generic

Here, a repository model that is suitable for audio data and implementation under ORDBMS will be discussed. The repository model can be considered as an extension of the image data repository model proposed in [38]. The proposed Audio Data Repository Model (ADRM) is a schema of 5 components: **A (ID, O, F, M, T)**, under an ORDB scheme, where:

- **ID :** unique identification that identifies an audio data in the database table
- **O :** an audio object that is used to store the physical audio data either as a Binary Large Object (BLOB), in the database, or as a BFILE , as an external file.
- **F :** set of low-level audio features that represent the audio object
- **M :** an object type used to describe the audio data using high-level metadata information of the audio description as per the classification given in the

audio data model (i.e. **M** represents the CPI and *conceptual* metadata as well as key(s) that are used in creating relationships with other tables). The *storage* metadata is embedded with in the audio object.

- **T :** temporal information associated with a given audio unit. It consists of the starting time of an audio unit.

"**O**" in the ADRM schema representation is the principal component of user's interest that refers to the audio document. The other elements like "**F"** and "**M**" are additional attributes that are meant to explain "**O"**. "**F**" in this work, represents feature vector representations that are extracted from an audio signal automatically. Basically, "**F"** is used in content-based audio similarity matching and identification. "**M**" is the feature that describes the high-level semantics of an audio. Some elements of "**M**" can be derived from the audio itself (e.g. the physical metadata) but mostly it needs human intervention. The component "**M**" is used in semantics based searching like keyword-based audio retrieval. The aforementioned components of audio data are used in image data repository model given in [38]. Whereas, the "**T"** component, which is unique for continuous media (audio and video), adds additional component (i.e., the temporal dimension). "**T"** will have a "NULL" value if the audio object "**O"** is considered as an indivisible unit (since the audio duration is embedded in "**O**") and its starting time is known, which is at 0:00. But, if "**O**" is structured into its sub components (e.g. speech) and if a random access of the audio clip is required capturing "**T**" is compulsory.

In [45], a temporal dimension is used to model a video data repository in addition to other components, which are relevant to model video at key-frame level. The temporal dimension, in our work, identifies itself in that it uses one element of the storage feature, "audio duration", instead of capturing the ending time of each audio unit. Thus, only capturing the starting time of audio is enough as ending time will be derived from the starting time and duration of a given audio unit.

Figure 3.2 shows the complete information that must be captured about audio data as per the representation proposed in the ADRM (details of the extraction process is given in section 5.1.2).

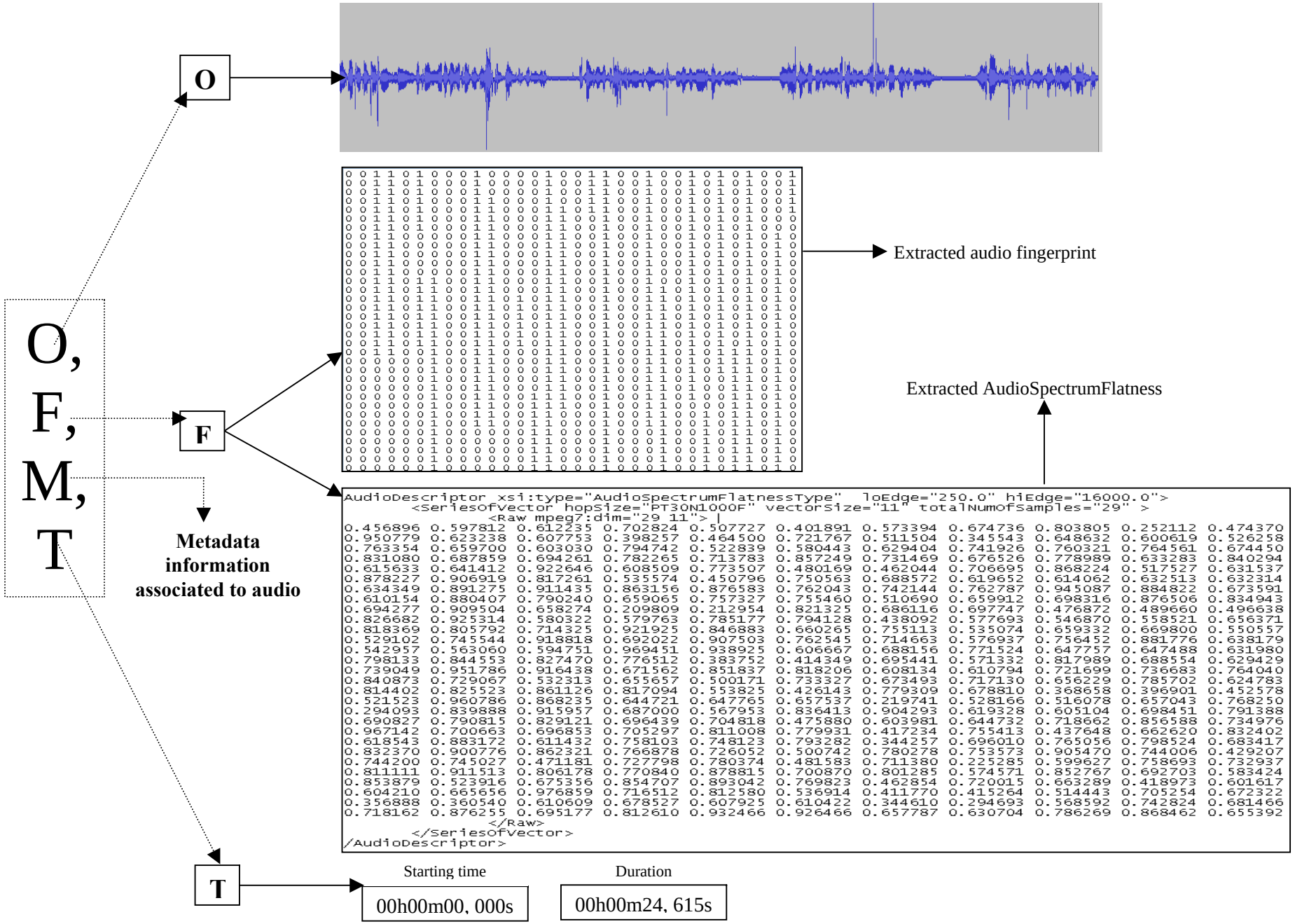


Figure 3-2: Content of audio representation based on the proposed repository model

3.2.1. Audio Data Repository Model: the case of speech

Speech is one subclass of audio data. Unlike, other types of audio categories, speech can be characterized by its hierarchical structure. In this subsection, the hierarchical units of speech and their way of representation in a database will be discussed. As per the structure given in the proposed data model, speech data can be organized into topics, sentences and words. As it has been discussed earlier, such classification enables users to retrieve only part of the speech in which they are interested in. The abovementioned ADRM can be applied to any generic audio data (music, speech or environmental sound). However, speech data needs further refinement as it can be represented in a hierarchical fashion. Thus, we have given speech extra consideration so as to show how each of the speech units can be modeled without loosing generality of the proposed repository model. As a result, the repository model of speech discussed in this subsection is derived from the generic repository model proposed previously. With this consideration, below we will give formal definitions for each speech unit starting from the "basic" unit - word.

Word (W) – is a basic speech unit where a set of words constitute a sentence. Although **word** is not a unit of user's interest in most cases, capturing its necessary information enables keyword-based audio retrieval and facilitates searching of a specific audio segment as per its importance to users. **Word** can refer to any of the words in a speech segment based on the application domain it is used. However, in this study, **Word** refers to keywords that are considered to be representative of a speech document by domain experts. **Word** is constructed using five components **W**(ID_w, O_w, F_w, M_w, T_w), where:

- ID_w : unique identification for a word
- O_w : refers to the word audio source or object
- F_w : low-level features extracted at word level.
- M_w : represents the alphanumeric data that is stored to represent the word, for instance the text that denotes the word.
- T_w : temporal information related to each word in a speech. This information can be managed in ASR easily as such systems capture the timing of each word in a speech.

Sentence(S) – a unit of speech which comprises a set of words. Sentence is a "basic" speech unit from user's requirement point of view that can convey meaningful information. It consists of five components: **S**(ID_s, O_s, F_s, M_s, T_s). All the elements of "**S"** share the definition given to "**W"** except that these components are captured at sentence level. Once a sentence that meets user's requirement is found, it is possible to broaden the result space by including sentences that precede/follow the sentence under consideration using the timing information they capture.

- ID_s : unique identification for a sentence
- O_s : refers to the sentence audio source or object
- F_s : refers to the low-level feature vector representation at sentence level
- M_s : denotes alphanumeric data that is stored to describe the sentence, it includes identification of all the words/keywords that constitute the sentence.
- T_s : temporal information related to each sentence in a speech. It holds the starting time of each sentence.

Topic (Tp) – is a subset of speech unit that contains sequence of sentences that disclose information about related issues. This unit is considered to express full information regarding a particular subject under consideration. **Topic** also has five components: **Tp** (ID_t, O_t, F_t, M_t, T_t). Once again, the components of this unit are almost identical to the previous ones except that the definitions are given at topic level. "**M"** represents the metadata information (e.g. description concerning the topic) that might be given at topic level as well as identification of all sentences that form the topic. Capturing sentence information enables to identify the "parent" topic encompassing the sentences.

Speech (Sp) - is a collection of one or more topics and is identified by five components: **Sp**(ID_{sp},O_{sp},F_{sp},M_{sp},T_{sp}). "**O_{sp}**" refers to the source speech itself. "**F_{sp}"** refers to the low-level representation extracted at speech level. The "**M_{sp}"** component holds the general description about the speech, the CPI metadata information as well as the identification of topics that makes the speech. The *CPI* metadata is inherited by all "child" units of speech (i.e. topics, sentences and words) because it is considered to be common to all sub units. "**T_{sp}**" holds the timing information of the speech.

3.2.2. Objects types of the Audio data repository model

Object types for each of the abovementioned components of the audio repository model are defined in this section. PL/SQL language is used to define these object types because it is one of the packages supplied by ORDBMS to extend database functionality and allow seamless access to SQL features. These definitions can be extended as per the requirement of the application domain in order to facilitate communication to other relational or object relational databases.

The object types are explained for generic audio as well as for the speech subclass. Note that the fundamental need to handle generic audio and speech separately stems from how the speech is considered during the audio retrieval process. If speech is considered to be atomic (a single audio unit), the generic repository model can be applied. On the other hand, if a specific segment of audio is to be accessed, which is comfortably possible in speech with its hierarchical classification, one can use the repository model proposed for the case of speech.

In the proposed speech repository model, we have considered word to be the basic and indivisible unit. Though it is possible to consider speech even at phone[7] level, the scope of this study limits itself to word level as we propose a multi-criteria audio retrieval that employs keyword-based retrieval approach.

The following SQL statements are used to create tables for general audio named **A** (if any audio is considered to be indivisible) and speech units (speech, topic, sentence and word identified as **Sp**, **Tp**, **S** and **W** respectively).

```
CREATE TABLE A  (Id Integer,
                 O Otype,
                 F Ftype,
                 A Atype,
                 T Ttype
                 );
```

[7] The smallest identifiable unit found in a stream of speech.

CREATE TABLE Sp (

Id_{sp} Integer,

O_{sp} OSP-Type,

F_{sp} FSP-Type,

M_{sp} MSP-Type,

T_{sp} TSP-Type,

);

CREATE TABLE Tp (

Id_t Integer,

O_t OT-Type,

F_t FT-Type

M_t MT-Type,

T_t TT-Type

);

CREATE TABLE S (

Id_s Integer,

O_s OS-Type,

F_s FS-Type,

M_s MS-Type,

T_s TS-Type,

);

CREATE TABLE W (

Id_w Integer,

O_w OW-Type,

F_w FW-Type,

M_w MW-Type,

T_w TW-Type,

);

The description of each of the object types is given below. The object type definition can be implemented under any ORDBMS (i.e. one shouldn't adhere to a specific product). But, for the sake of demonstration, we have used PL/SQL under Oracle9i.

3.2.2.1. Object types for the generic audio

Otype

Otype object type contains the actual audio data or a reference to it. It includes all the storage attributes that are associated to audio, which can be accessed through the audio source. It is mainly used in supporting the storage and management of audio data. The **Otype** object type and the minimum collection of attributes included in it are defined as follows:

```
CREATE OR REPLACE TYPE Otype
AS OBJECT
(
        -- ATTRIBUTES
        AudID INTEGER,
        AudioSource ORDSYS.ORDAUDIO,
        Description VARCHAR2(4000),
        Format VARCHAR2(400),
        MimeType VARCHAR2(4000),
-- AUDIO RELATED ATTRIBUTES
        Encoding VARCHAR2(400),
        NumberOfChannels INTEGER,
        SamplingRate INTEGER,
        SampleSize INTEGER,
        CompressionType VARCHAR2(400),
        AudioDuration INTEGER,
        --Methods
-- The following presents only some of the methods used for audio data manipulation.

MEMBER PROCEDURE setSource(source_type IN VARCHAR2,source_location IN
                                VARCHAR2, source_name IN VARCHAR2),
MEMBER FUNCTION getSource( ) RETURN VARCHAR2,
MEMBER PROCEDURE setDescription(user_description IN VARCHAR2),
MEMBER FUNCTION getDescription( ) RETURN VARCHAR2,
```

```
MEMBER PROCEDURE setFormat(knownformat IN VARCHAR2),
MEMBER FUNCTION getFormat( ) RETURN VARCHAR2,
MEMBER PROCEDURE setMimeType(mime IN VARCHAR2),
MEMBER FUNCTION getMimeType( ) RETURN VARCHAR2,
                        …
);
```

Where,

Description	: description of the audio object.
Source	: the source where the audio data is to be found.
Format	: format in which the audio data is stored.
MimeType	: MIME type information.
Encoding	: encoding type of the audio data.
numberOfChannels	: number of audio channels in the audio data.
samplingRate	: rate (in Hz) at which the audio data was recorded.
sampleSize	: sample width or number of samples of audio in the data.
compressionType	: compression type of the audio data.
audioDuration	: total duration of the audio data stored.

Ftype

Ftype is an object type that stores audio feature vector representation, which can be extracted automatically. **Ftype** is used in content-based audio retrieval as this object is involved in audio similarity matching between different audios. **Ftype** can be defined as follows:

```
CREATE OR REPLACE TYPE Ftype
AS OBJECT
(
        F CLOB
           ...
-- Methods

-- Only some of the methods used are listed here
```

STATIC FUNCTION initialize () RETURN Ftype,

STATIC FUNCTION CalcDist() RETURN FLOAT,
);

Where,

F is the low-level feature(s) that corresponds to the type of audio class under consideration. It can be extended as per the application requirement as the various audio classes necessitate different audio features.

Mtype

Mtype represents the high-level semantics feature of a given audio signal. This information can be captured either in a fully annotated form automatically (e.g. from ASR) or manually as the case of domain dependent keywords (e.g. conceptual metadata) and CPI metadata. The definition of **Mtype** object type is given as follows.

CREATE OR REPLACE TYPE **Mtype**

AS OBJECT
(
KW : VARCHAR2 (50),
CPI : VARCHAR2 (50),
Key : VARCHAR2 (50),
…
-- Attributes of **Mtype** can be extended based on the application requirement.
--Methods

-- Only some of the methods are presented here. The methods can be extended as per the need of the application under consideration.

STATIC FUNCTION enterCPImdata()

STATIC FUNCTION getCPImdata () RETURN VARCHAR2 (100)

STATIC FUNCTION enterKW()

```
STATIC FUNCTION getKW () RETURN VARCHAR2 (100)
        …
);
```

Where,

KW : refers to list of keywords that can serve as metadata that give conceptual information regarding the audio segment.

CPI : represents all the information concerning the creation and production of audio.

Key : an attribute used in communication between other relational and object relational tables that are directly or indirectly related to the audio table.

Ttype

This object type stores the time information of a given audio document. As it is seen above, the audio duration is embedded within the audio data. Thus only the starting time will be captured by this object type as ending time can be calculated from the starting time and duration. Its definition is presented as follows.

```
CREATE OR REPALCE TYPE Ttype
AS OBJECT
(
STime FLOAT,
--Method
MEMBER FUNCTION setSTime ()
MEMBER FUNCTION getSTime() RETURN FLOAT,
);
```

Where,

STime: holds the starting time of an audio document.

3.2.2.2. Object types for the speech sub-class

The reason for dealing with the speech sub-class independently is that it is possible to structure speech into its hierarchical units. Though most of the definitions are similar to the above mentioned ones, the following object type definitions will explain the relationship among the different speech units. Most of the methods are left intentionally because they are similar to those

defined for the generic audio. The object type definitions that are used in each speech units are defined below.

Word level object definitions

The word level object definitions explain all the objects that are held in creating a word table by considering word as an indivisible speech unit. Its definitions are almost similar to that of the generic audio. However, for the sake of completeness, they are included here.

OW_Type

CREATE OR REPLACE TYPE **OW_Ttype**
AS OBJECT
(
WSource : ORDSYS.ORDAUDIO,
);
Where,
WSource : refers to the audio source of the word (audio) data.

FW_Type

CREATE OR REPLACE TYPE **FW_Type**
AS OBJECT
(
F : CLOB,
);
Where,
F: denotes the feature(s) vector representation of the word which is/are used in similarity matching between two words (audio).

MW_Type

CREATE OR REPLACE TYPE **MW_Type**
AS OBJECT

(

Mdata : VARCHAR2(2000),

);

Where,

Mdata represents any alphanumeric data associated with the word (e.g. its text)

TW_Type

CREATE OR REPLACE TYPE **TW_Type**

AS OBJECT

(

ST: VARCHAR2 (2000),

);

Where,

ST refers to the starting time of each word.

Sentence level Object definitions

The sentence level object type definition is also identical to that of the word. **FS_Type** and **TS_Type** capture the low-level feature and the starting time of a sentence respectively. The object type definitions of **OS_Type** and **MS_Type** are shown below.

OS_Type

CREATE OR REPLACE TYPE **OS_Type** AS OBJECT

(

SSource : ORDSYS.ORDAUDIO,

);

Where:

SSource : The audio source that refers to the sentence itself. It is used in playing a sentence if a user is interested to listen to a sentence that contains a given keyword.

MS_Type

CREATE OR REPLACE TYPE **MS_Type** AS OBJECT

(

KWID VARCHAR2(25),

Key VARCHAR2(50),

);

Where,

KWID: refers to the identification of all words (keywords) that are contained in the sentence. Capturing this information enables an audio retrieval system to decide which sentences are relevant to user queries (i.e. the more keywords a sentence contain the more relevant it is).

Key : allows to create relation with other relational or object relational tables a sentence level speech is referring to(e.g. If an image description is considered, a single sentence might tell a valuable information regarding a salient object of an image). In such cases, identification of the salient object can be related to the "key" attribute).

Topic level object definitions

OT_Type, FT_Type and TT_Type have the same object definition as that of **OS_Type, FS_Type** and **TS_Type** respectively except that they are made at topic level. The following defines the object type **MT_Type**.

MT_Type

CREATE OR REPLACE TYPE **MT_Type** AS OBJECT

(

ID VARCHAR2(25),

Key VARCHAR2(50),

);

Where,

ID : refers to the list of sentence identifications that constitute the topic.

Key: enables to create relationships between the topic table and other tables.

Speech level object definitions

All the object definitions are the same to that of topic except that they refer to speech components. **MSP_Type** contains all the *CPI* metadata that will be inherited by all speech units. It also includes identification of all topics which construct the speech and keys that allow it to communicate with other external tables.

Note that, for all sentences, topics and speeches, identifications of their constituent words, sentences and topics respectively are captured. Firstly, such information is used in determining the relevance of a speech unit. Secondly, the above mentioned speech units are stored in their entirety to enable playing of a relevant speech segment without trying to construct "parent" units from their "child" units.

The major reason for preferring to store the sentence/topic/speech themselves instead of composing them from words/sentences/topics respectively is that doing so becomes costly as it is a time-intensive task which creates bottleneck in retrieval's performance. If storage is an issue, the source object types (**OS_Type, OT_Type ,OSP_Type)** can be "NULL" and will be constructed from their constituent units, but at the expense of performance.

3.3. Summary

Audio is a complex, data-intensive and content rich media that can convey more information than using mere text. With such unique characteristics, audio has been an active research area for the last two to three decades. However, audio content analysis especially with respect to providing proper data and repository models are still in their preliminary stage. Thus, there is an imperative need to have a model that captures the inherent characteristics of audio data and characterizes audio representation in a database so as to make it effectively and efficiently searchable. The aim of this study is in accordance with this necessity. In this chapter, we proposed an audio model that incorporates both the low-level and high-level features of audio based on MPEG-7 description scheme. In addition, we proposed an audio repository model that can be considered as a mirror reflection of the data model that facilitates a convenient storage of audio data in a DBMS. We have grouped audio data repository model in two groups: generic and speech. For the

generic case, we have proposed a generic data repository model that can be applicable to all type of audio data if and only if a given audio document is assumed to be an indivisible audio unit. Since speech has a hierarchical representation, we have also provided a repository model that takes these structural speech units into account. In this work, we have taken advantage of ORDBMS in storing and representing audio data, a task which has been a challenge in traditional DBMSs. Though we defined the object definitions in connection with Oracle9i, they can be used in any ORDBMS. Both the proposed models are general in that they can be applied in an unconstrained domain.

4. Audio Data Management for Medical Application (ADMMA)

4.1. Application domain used

In the previous chapter, an audio data and repository model which enables us to capture both the low-level and high-level features of an audio data is proposed. The proposed models can be applied to any audio application domains. But, for the sake of demonstration, we have selected a medical application and developed a prototype –ADMMA- to show their practicability. This prototype limits itself to only two classes of sounds that are common in the medical environments: heart sounds and speech-based medical image descriptions.

In the first attempt, we have tried to manage an audio signal from the environmental sounds category which is a human heart sound. Among the various audio-based medical diagnosis practices, analysis of human heart sounds (even if done by humans) is popular in acquainting physicians with health problems a patient has. Currently, physicians listen to patient's heart sounds and make decisions based on their subjective impression. But, the main problem is, most of these sounds especially those categorized as "abnormal" are hardly noticeable by most of the physicians other than those who are specialist in the specific area. Such problems call for systems that support the identification of those sounds instead of being reliant solely on domain specialists. This prototype can be used as a starting point for using audio retrieval systems to assist physicians in differentiating between the different heart sounds by simply giving input heart sound and querying the system to retrieve similar sounds and (if needed) related information. Furthermore, we have demonstrated that heart sounds can be retrieved based on the high-level features in which they are described. Such systems can be convenient to learn to diagnose heart sounds and to develop teaching aids for auscultation training.

In the second case, a recorded speech of medical image description is considered to illustrate audio retrieval techniques at high-level features, structured based on the proposed audio repository model; these techniques can facilitate communication among the concerned parties (e.g. physicians) as current way of communication among physicians is through a written text and/or medical images. This prototype can also be used as a baseline to make such communications in spoken documents as speech is natural, easier and faster to be used in tasks like image descriptions.

All information regarding the above mentioned audio data is stored as MPEG-7 conforming descriptions. Then, content-based and keyword-based audio retrieval are applied using low-level and high-level features respectively. Figure 4-1, depicts the higher level architecture of the system, which shows the flow of audio retrieval. First, audio features will be extracted from an input audio signal. These features are then described or represented based on MPEG-7 description schemes and stored in the database. When a user queries the system, depending on the type of query (content or keyword-based) the query processing will either extract audio features and search for similar sounds in the database(represented as storage in the figure) or directly retrieve sounds from the database that fulfill user requirements.

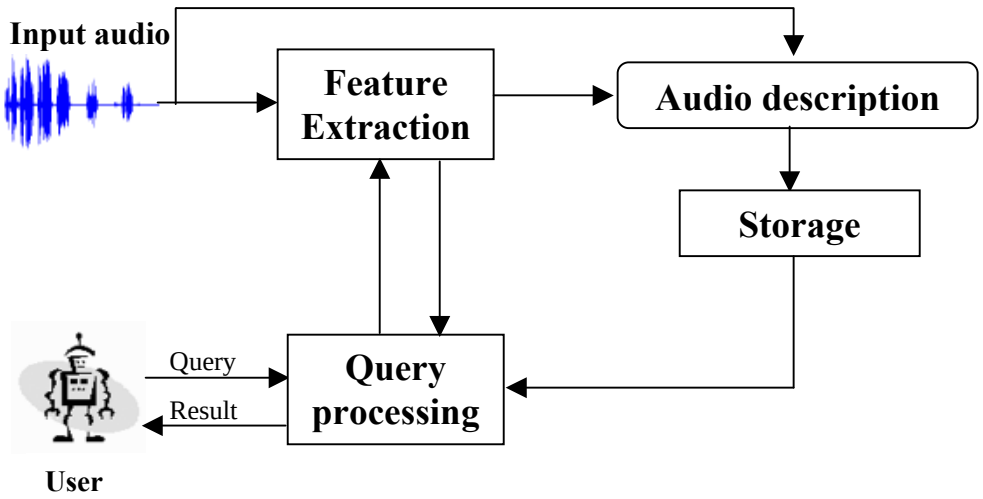


Figure 4-1: High-level architecture of ADMMA

We have chosen ORDBMS for managing the audio data in our prototype. The reason for choosing ORDBMS is that, such systems allow integrating tools that can be used to manage audio together with its related alphanumeric data (see section 3.2.). ADMMA provides an interface that allows retrieving heart sounds using multi-criteria (using both the high-level and low-level features of audio data) query formations. To store and access audio data, we have used the accessibility provided by Oracle intermedia, which is a component integrated with Oracle for the purpose of multimedia manipulation. The following section gives some insight concerning Oracle intermedia.

4.2. Oracle interMedia

Oracle interMedia is a feature that enables Oracle9i to store, manage, and retrieve multimedia information in an integrated fashion with other enterprise information. With Oracle interMedia, multimedia data can be managed as easily as standard attribute data. Some of the pluses of interMedia are listed below [65].

Oracle intermedia:

- Provides the means to add audio, image, video, and other heterogeneous media columns or objects to existing tables and retrieve them back.
- Manages industry-standard audio data stored in different formats, extracts metadata information from these formats, and stores it as attributes of the Oracle interMedia audio object (ORDAudio) within Oracle9*i* table space.
- Provides considerable flexibility in the storage of the actual media as well as storage of the media outside the database by integrating it with the archival storage.
- Can be accessible to applications through both relational and object interfaces. Database applications written in Java, C++, or others can interact with *inter*Media through modern class library interfaces, for instance using PL/SQL[66].

4.3. Java Media Framework

Java Media Framework (JMF) provides a unified architecture and messaging protocol for managing audio data and time-based media data in general. With JMF, one can easily create applets and applications that present, capture, manipulate, and store time-based media. The framework enables developers and technology providers to perform custom processing of raw media data and seamlessly extend JMF to support additional content types and formats, optimize handling of supported formats, and create new presentation mechanisms. In this particular prototype, among the various facilities JMF can provide, our focus is on the presentation of audio data. To present time-based media, JMF provides a player in which playback can be controlled programmatically or through a control-panel component. JMF also offers a Java Bean known as MediaPlayer that encapsulates a JMF player to provide an easy way to present media from an applet or application. Owing to such advantages of JMF, we integrate our application with JMF in order to playback an example audio that the user is interested in searching as well as the query results that are assumed to be relevant to user queries.

4.4. General Architecture of ADMMA

ADMMA is developed using Java and the JDBC interface for Oracle9i. It uses the audio data and repository model proposed in this thesis to store audio data. Java is chosen as a development environment since it enables to create a platform independent application. In addition, most techniques used in the prototype are mainly supported by Java. For instance, interMedia Java Classes that describe OrdAudio object type which are used in supporting the storage and management of audio data are provided by Java. Additionally, Java offers versatile APIs in order to provide a unified protocol for managing the acquisition and delivery of audio data for instance, Java Media Framework (JMF). A simplified architecture of ADMMA is depicted in Figure 4-2 and an overview of the layers will be given on the subsequent sub sections.

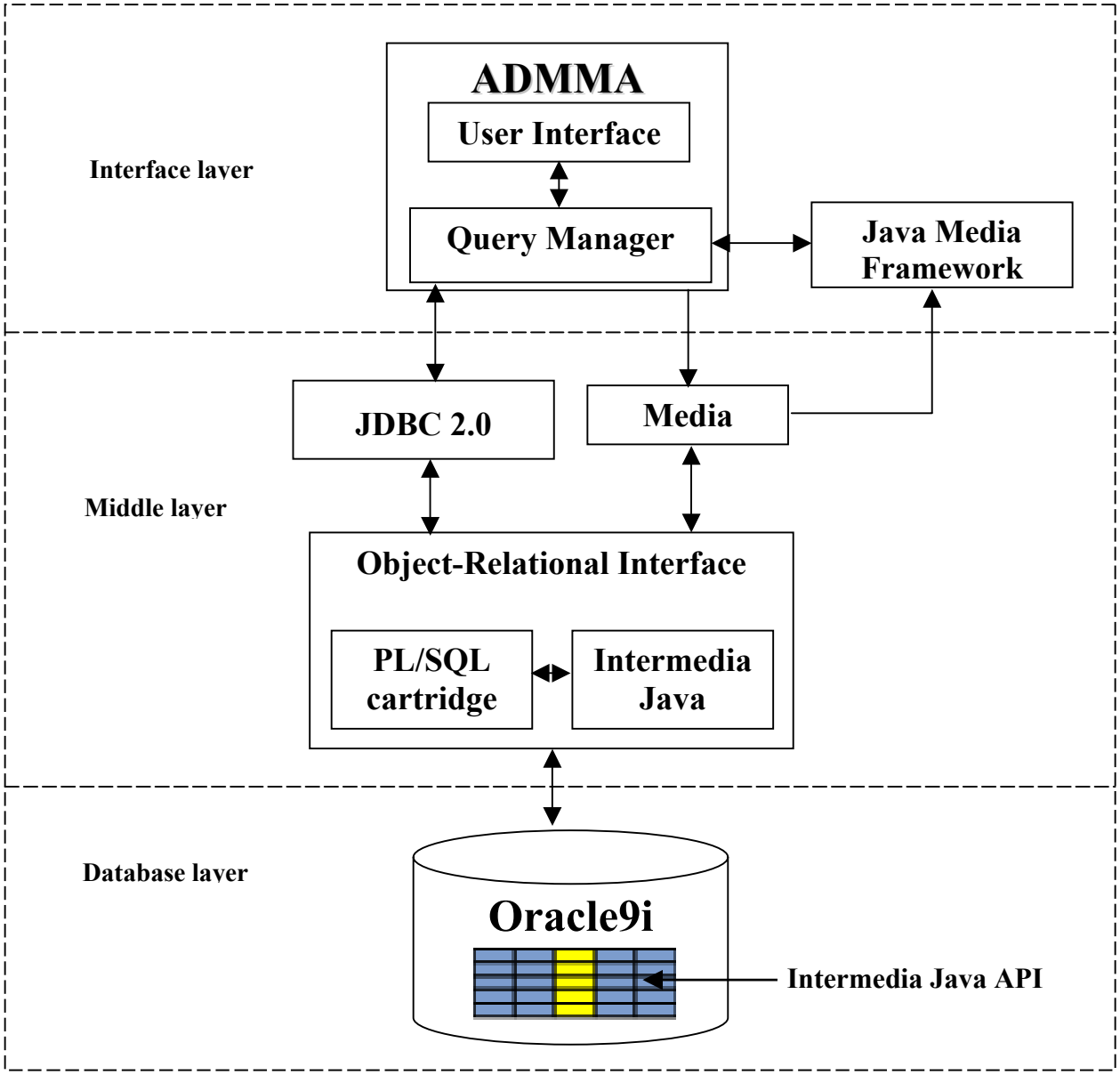


Figure 4-2: Simplified ADMMA architecture

Interface layer

This layer provides mechanisms by which users can interact with the system and portrays interfaces to support audio content search and display functionalities. The system runs as a standalone process but it can also be made accessible through internet by extending it to run as a Java applet. The user interface mainly has two components: the audio entry and audio retrieval. In both cases, JMF is used to playback audio and this enables users give subjective evaluation on the similarity of two or more audio signals. Brief description of the audio entry and retrieval interfaces will be given in section 4.7.
In general, the interface layer provides the following functionalities.

- Interface to new audio data entry
- Interface to enable users to select sample audio data
- Interface to let users search audio that resembles a given sample audio data
- Interface to search for audio files in the database based on audio name, description, format, encoding, number of channels, sample size etc.
- Interface to display Audio Search Results

Middle layer

Java utility programs are implemented using JDBC and intermedia Java API to load and retrieve audio data to/from Oracle9i database. PL/SQL through ORDAudio object type is used to access Oracle interMedia audio content using JDBC to access the structured (relational) and unstructured (media files) data from database. Oracle *inter*Media Java Classes makes it possible for JDBC result sets to include both traditional relational data and *inter*Media media objects (OrdAudio in our case). Such facilities allow our application to easily select and operate on a result sets that contain *inter*Media columns plus other relational data.

Database layer

The database layer comprises Oracle9i database management system. This layer interfaces with PL/SQL and Java intermedia classes. PL/SQL is used to load the audio data from external audio files in to a table of an Oracle9i database containing intermedia object type column. Java classes

are used to write Java applications using intermedia objects. They also enable access to *inter*Media object attributes and invocation of *inter*Media object methods. By taking advantage of these two components, the Oracle database is made to hold the audio data along with the traditional data. Figure 4-3 illustrates the set of information that is retained under the ADMMA database.

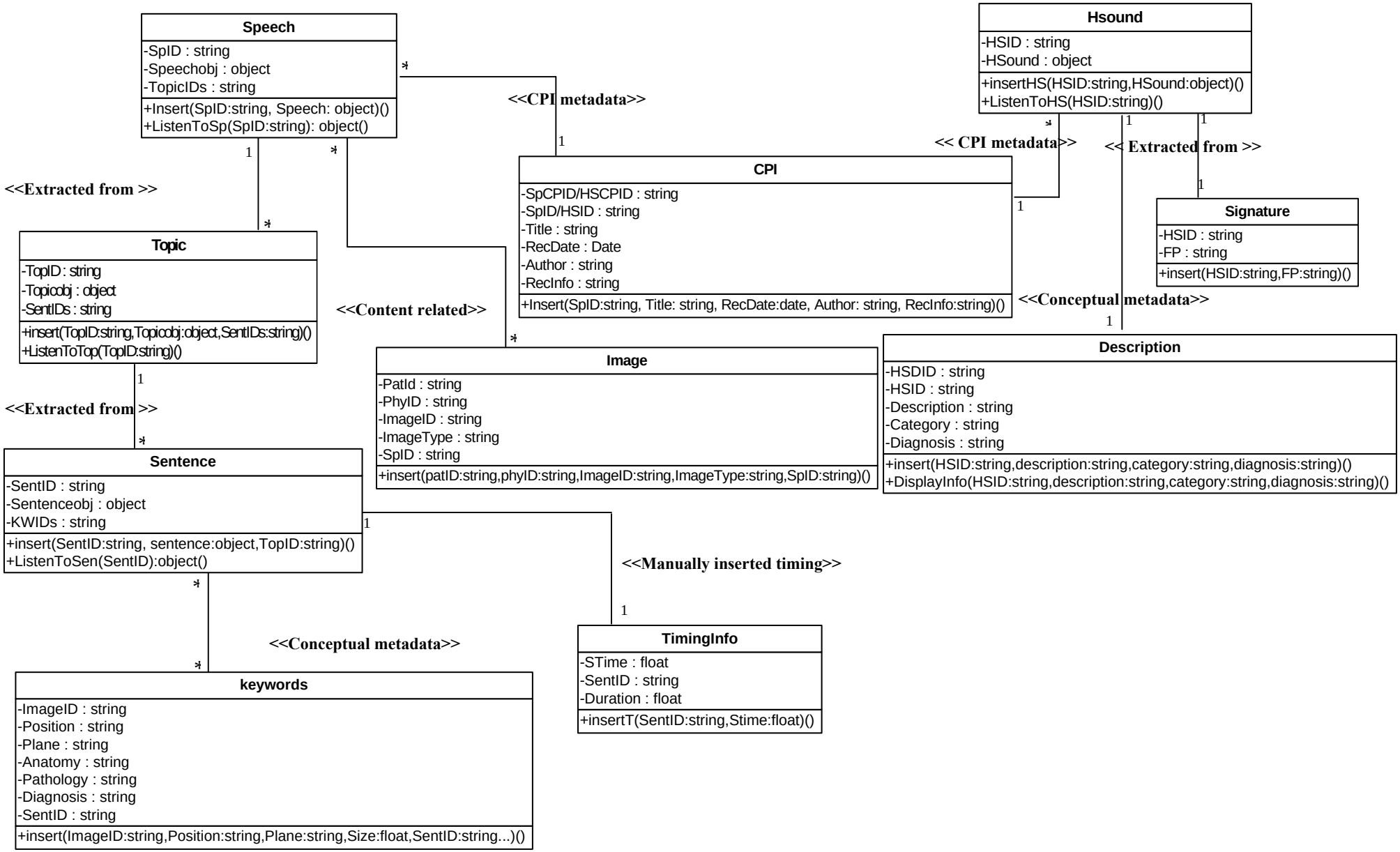


Figure 4-3: Class diagram of ADMMA in UML representation

4.5. Components of ADMMA

This sub-section presents some of the components of ADMMA that are stored in the database in compliance to items given in the proposed models. As it is discussed previously, the proposed models can be applied in generic audio applications under any ORDB schema. To demonstrate their applicability, we have presented the sample data that we used in our prototype under the Oracle9i framework. Our database keeps information so as to make audio data searching easier based on the proposed multi-criteria query formation. For the sake of simplicity, we have shown the two audio-based medical applications in to two categories: Speech based medical image description and heart sound.

Speech based medical image description

Image *(**ImageID**, ImageType, SpID, PatID, PhyID)*; is a table that contains information related to the medical image that is to be described. It contains the image identification, the type of medical image, the speech to which the image description belongs to as well as the patient and physician identifications. The PatID and PhyID are used as primary keys in their respective tables and as a foreign key in the Image table to allow manipulation of patient and physician information. An image is uniquely identified by its ImageID, which can also be used as a foreign key in other tables.

Patient (***PID***, *First name, Last name, Age, Address, MHistory*); is a table containing patient information. This table is uniquely identified by its PID.

Physician (***PHID***, *First name, Last name, Address, Specialization*); is a table containing physician information, which is uniquely identified by its PHID.

ImageDesc *(**SpID**, ImageID, Position, Plane, Anatomy, Pathology, Diagnosis, **SentID**)*; is a table that holds information related to the speech based description of an image. It stores the position, plane, diagnosis, anatomic and pathologic information as well as additional keywords that are applicable to the type of image under consideration. It also contains the speech and sentence identifications which are used to uniquely identify the speech.

Note that in this prototype, a speech is made to contain a single topic about the image it describes. Thus, the topic table definition is not considered: instead, TopID and SpID are referring to speech identification in this particular case. Such an audio table can answer a query like "*List all audio that tell about chest x-ray descriptions taken at posterior-anterior position which are believed to be a lung cancer*".

Sentence *(**SentID**, sentence, TopID, KWIDs, STime)*; is a table that contains information concerning a single sentence that is identified by SentID. TopID(in this case similar to SpID) is used to locate to which topic or speech a sentence belongs to. The timing information is captured using STime, which captures the starting time of a sentence so as to enable listening to a single sentence or a sentence together with its preceding and following sentences. This table can be used for a query like "*List sentences that talk about uretral stone in an x-ray image*".

SpCPI(SpCPID*, SpID, Title, RecDate, Author, RecInfo)*; is a table holding information that deals with the creation and production of a speech. It is uniquely identified by SpCPID(HSCPID for the case of heart sounds). The general description about the speech/heart sound is stored as Title. The author and other information regarding the recording (e.g. studio/live recording) are identified by RecInfo. For example, from this table, it is possible to answer a query like "*Select audio recordings about Obstetric Ultrasound that are recorded after 1/1/1999 with a studio quality*".

Heart sounds

HeartSound(HSID*, HeartSound, FP)*; a table holding information that needs to be captured in relation to a given heart sound. HSID is a unique identifier that identifies a heart sound. The signature (fingerprint) that is stored in the database and used in identifying a given heart sound is captured by FP. With this table one can answer a query like "*Search all sounds that sounds like an example abnormal heart sound.*"

HSdesc(HSDID*, HSID, Description, Category, diagnosis)*; a table that contains the high-level description of heart sounds. It consists of the description of the sound, the category (e.g. normal or abnormal) and the possible diagnosis that are considered to be causes for a given heart sound especially for those sounds that fall under the "abnormal" category. This table can answer a query

like "*List abnormal sounds that are identified as 'accentuated 1st sound' together with their possible causes*".

Audio tables in the database have the following structure:

A (***ID***, *O, F, M (ImageID, TopicIDs, SpCPID*), *T*) – is used for the case of image description at speech level. Attributes like ImageID, TopicID, SpCPID enables the audio table to communicate with external tables, for instance, to retrieve image information, CP information etc.

A (***ID***, *O, F, M* (*HSDID, PID, PHID*), *T*) – is used for the heart sounds scenario. HSDID, PID and PHID are used to relate the audio table with other table(s) that store heart sound description, patient and physician information.

4.6. Retrieval Approaches used in ADMMA

Audio retrieval approaches employed in our prototype are content-based and keyword-based audio retrievals. The following subsections will briefly discuss these retrieval approaches.

4.6.1. Content-based Audio Retrieval

In this section, the content-based audio retrieval approach, specifically the query-by-example approach, used in this work will be presented. As it is discussed in section 2.2.2, query-by-example is usually done in the following ways: first, features are extracted from an example and all sample audios in a database. Second, the distances between the feature vectors of the example and the database samples are estimated using a certain distance metric. Finally, database samples having the shortest distance to the example are retrieved [66]. Thus, in our prototype, a compact representation of feature vectors is extracted from each audio sample and stored in a database. Typically, we used an audio fingerprint as a low-level representation of an audio data after thorough investigation of low-level features (see the next chapter) that best suit for the type of audio considered in this prototype. Once again, the fingerprint of each query sample is extracted and compared to that of samples in a database to determine the similarity among them.

Content-based audio retrieval is based on similarity instead of exact match between queries and database items. It has been a great challenge to define what audio similarity really is. Even

humans have great difficulties in describing what makes two pieces of audio, for instance music, similar. Here, the similarity of two audios is defined as the proximity of their feature vector representations in a feature space. Since the relevance of retrieved results is judged by human beings, the main requirement of similarity measurement is that calculated similarity values should conform to human judgment. Even though it is not possible to give a general definition for audio similarity, similarity measures intend to capture what human listeners hear when listening to a given sound. During content- based audio retrieval, obviously, to decide “close” or “far”, the similarity measure plays an equally important role as the original feature space.

In this prototype, the fingerprint extraction algorithm used works in such a way that audio fingerprints are represented as a series of bit strings. Thus, the number of bit errors between two feature vectors determines the similarity/dissimilarity between their respective audio signals. The distance measure employed in this prototype is known as *Hamming Distance*. Hamming distance was originally conceived for detection and correction of errors in digital communication. It is simply defined as the number of bits that are different between two bit vectors.

A simple algorithm that calculates the Hamming Distance, which in turn is measured in terms of bit error rates (i.e. the smaller the number of bit error rates the more similar the audio signals are), is given in [67]:

```
int hamdist(int x, int y)
{
  int dist = 0,
  int val = x^y;
    while (val)
  {
    dist++;
    val = val - 1
  }
  return dist;
}
```

4.6.2. Keyword-based Audio Retrieval

In the keyword-based retrieval, domain dependent keywords are selected with the support of domain experts and other metadata information that are associated with the media are extracted

from the audio object itself. We used the traditional IR techniques to retrieve an audio document that contains a given keyword. We have demonstrated the keyword-based retrieval for each of the sounds (speech-based image description and heart sounds) considered in our prototype.

4.7. ADMMA Interfaces

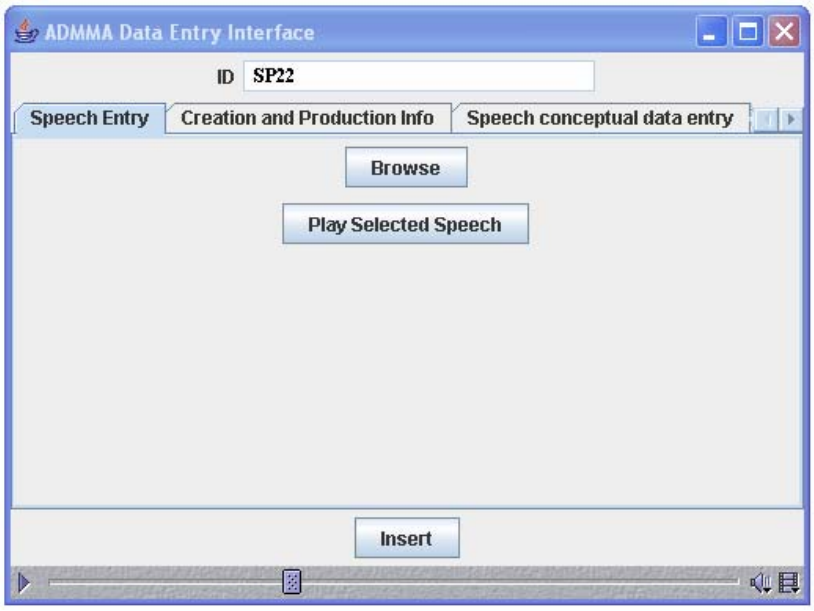
This section mainly deals with presenting the various interfaces that are used in the ADMMA prototype to enable audio data entry and search.

4.7.1. Audio entry interface

ADMMA audio entry interface is concerned with providing a way to enter new audio data in to the database. Some of the tasks in the data entry are done programmatically while some are made possible through the use of off-the-shelf tools. In order to incorporate metadata information other than those that are set in the media, we used "IBM Multimedia Annotation Tool", which is a tool that assists in annotating media files with MPEG-7 metadata. By recognizing the key role that MPEG-7 can play in audio feature representation, we have incorporated it as a content descriptor interface for consistent representation of all metadata information related to audio, which also include keyword as its *conceptual* metadata. Some of the jobs that are done programmatically are extraction of audio fingerprints and segmentation of input speech into sentences. Java GUI (Graphical User Interface) is used to implement the front end screens. This prototype currently supports four interfaces to let users enter audio data: Audio source entry, Encoding schemes, *CPI* and *Conceptual* metadata entry.

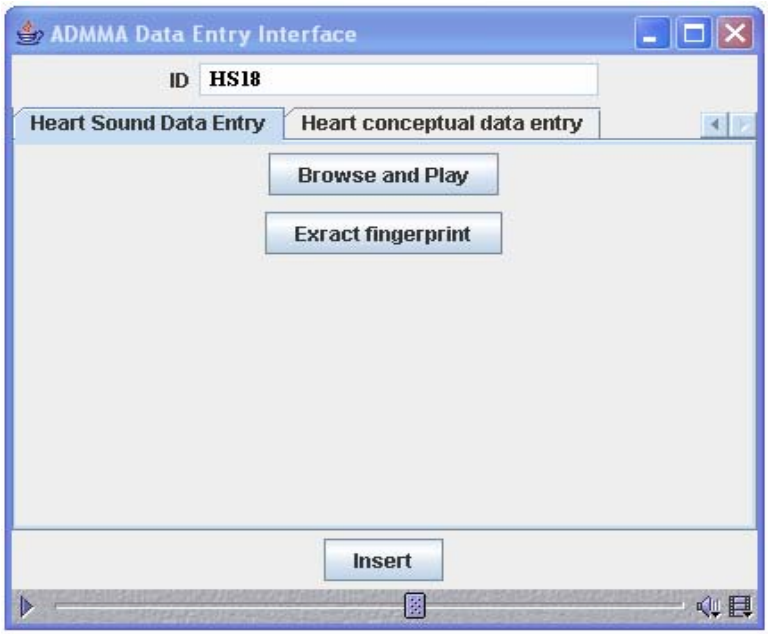
4.7.1.1. Audio Data Entry

Audio data entry lets users select an audio file and enter it into the Oracle9i database that is used for storing audio information. It also provides a means with which one can extract the low-level signature/fingerprint that is to be used in similarity checking. Figure 4-4(a) and (b) shows the audio source entry interface. Figure 4-4(a) is the speech based medical image description entry interface. This interface provides users with four panels that are used to capture different information. The speech entry panel enables users to select (browse) and play the audio (speech) to be entered. If the user decides to insert the speech he/she listened to, pressing the insert menu enables to store the speech data into the "O" component of the audio.



(a) Speech data entry interface for medical image description

Figure 4-4(b) shows the screen shot of the Heart sound entry interface. Similar to that of speech based image description, this interface allows users to browse and play selected audio. If the user is interested to store the audio data(heart sound), he/she can use the "extract and store fingerprint" button to extract the fingerprint of the heart sound and store the source data and the fingerprint in the "O" and "F" component of the audio table respectively up on clicking of the "insert" button.



(b) Heart sound entry interface

Figure 4-4: A screen shot of audio data entry interface

4.7.1.2. Audio encoding and CPI entry interface

The encoding as well as *CPI* that are associated with the audio data are entered through the audio encoding and CPI entry interface. Such metadata is used in queries that target encoding and production information. Figure 4-5 shows the screen shot of this interface. Basically, most of the encoding information are attached to the audio itself (sampling rate, number of channels, sample size, format etc) and are available with the source automatically. But, some of them need human intervention and manual entry. Such information and CPI metadata are entered and stored in the "M" component of the audio database when the user clicks "insert" button.

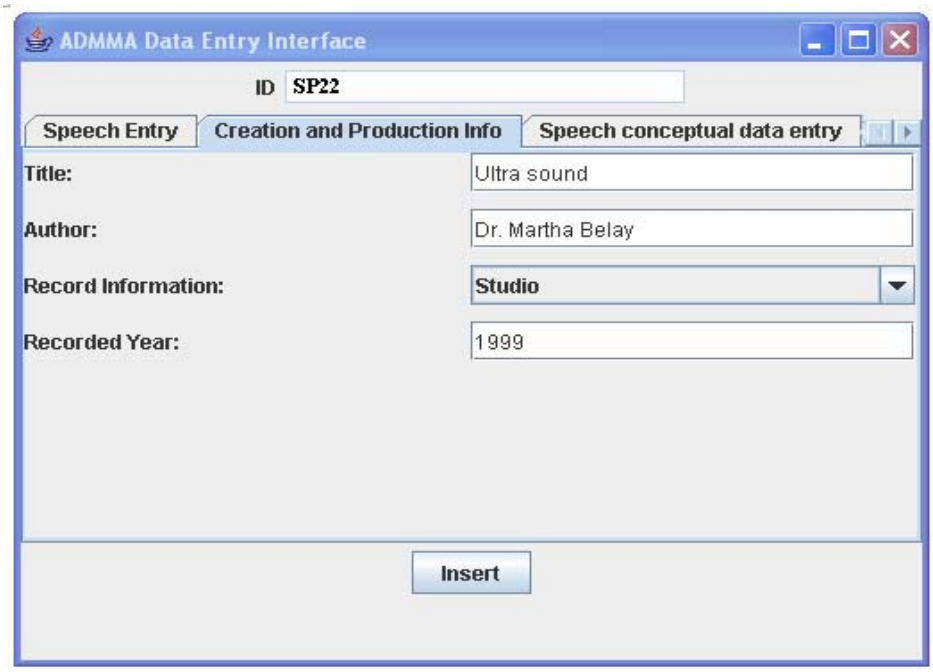
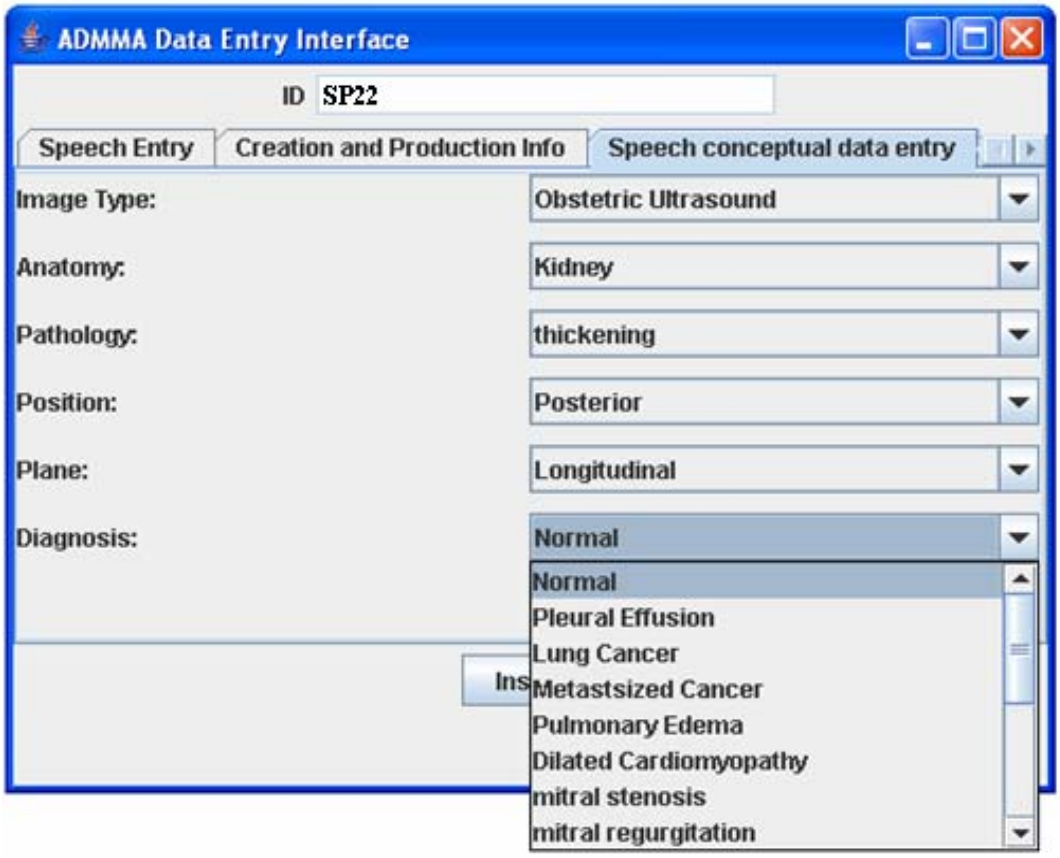


Figure 4-5: A screen shot of audio CPI interface

4.7.1.3. Audio Conceptual Data Entry Interface

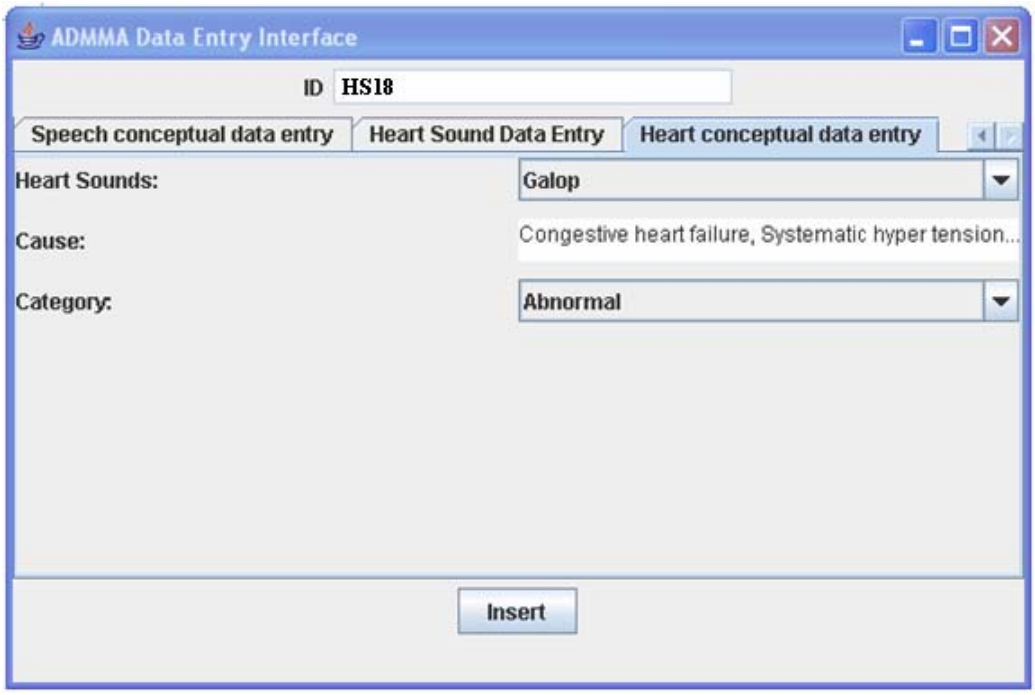
Audio conceptual data entry interface is an interface that enables users to enter high-level information of a given audio. This information is related to contents of audio data that are used in keyword-based audio retrieval. The audio conceptual data entry interface limits users only to select from list of values instead of entering free texts. The reason is that this interface provides exhaustive list of possible application-dependent values for each application-dependent keywords (e.g. Image type, anatomy, pathology etc), that are believed (by domain experts) to be universal in medical images.

Users are allowed to select keyword values that are used in speech based medical image description from the list provided. Clicking the "*insert*" button initiates the task of storing all selected values in the "M" component of the audio database.



(a) Speech conceptual data entry

Figure 4-6(a) shows the screen shot of conceptual data entry interface for speech based medical image description. Figure 4-6(b) depicts the interface used in capturing semantic/conceptual data of heart sounds. It provides exhaustive list of the various heart sounds (e.g. Gallop, Holosystolic murmur, Late systolic murmur etc.) together with their associated causes (especially for abnormal heart sounds). Users are allowed to enter free text for cause since abnormal heart sounds can be caused by different health problems. Similar to the image description, when the "insert" button is clicked, these information will be stored in the "M" component of the audio database.



(b) Heart sounds conceptual data entry

Figure 4-6: Screenshot of audio conceptual data entry interface- (a) and (b)

4.7.2. Audio Retrieval Interface

This interface is responsible for rendering query results to users. It offers a multi-criteria query formation to let users search audio either by giving example audio or domain dependent keywords and provides users with an interface to browse returned audio objects.

4.7.2.1. Query-by-example audio retrieval interface

Query-by-example audio retrieval interface allows users to select an example audio file. It, then, computes similarity and gives list of audios that are relevant to a query along with their associated metadata. It provides users with an interface to browse and listen to example audio as well as responses that are believed to be relevant to a given query. Figure 4-7 shows the screenshot of query-by-example audio retrieval interface. When a user clicks “browse”, the list of available audio data that are used as example audios will be displayed and the selected audio will start to play. If the user is interested to search for a similar sound he/she can click the “search similar sound” button. The system then returns list of similar sounds through a result table. To playback the result, the user will select the result and click on “play selected Audio” button.

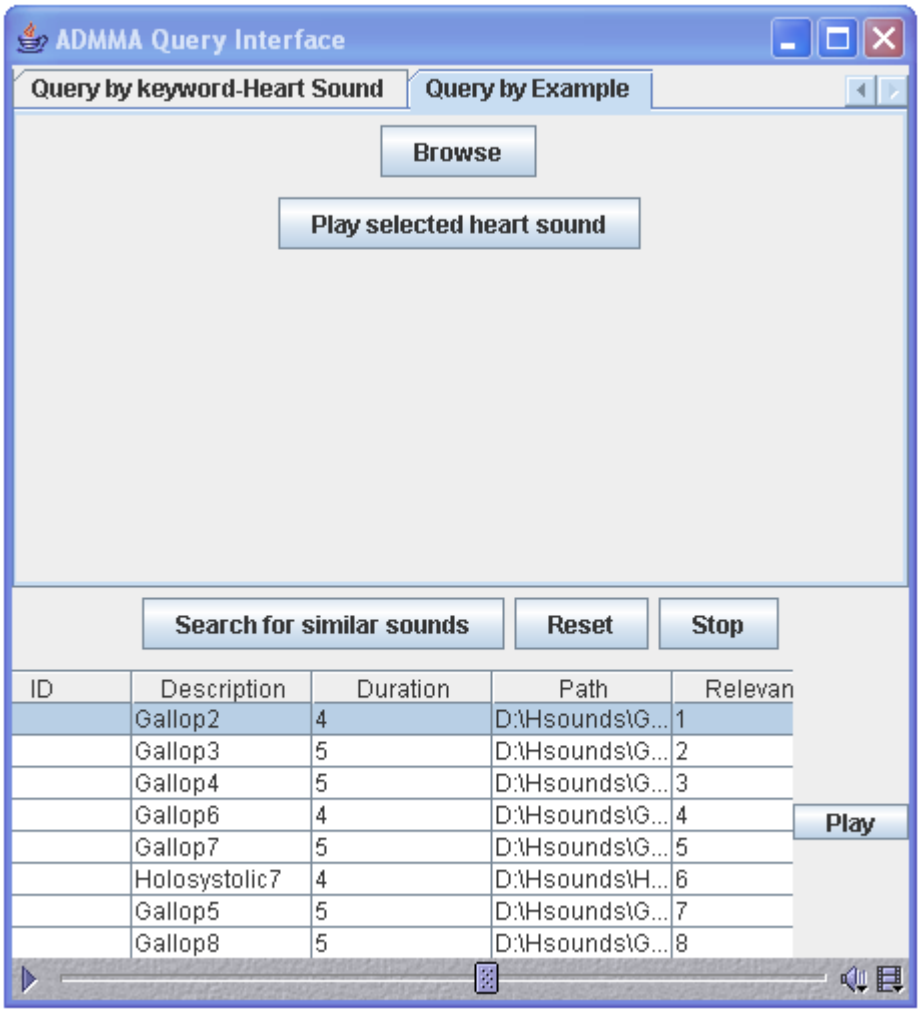


Figure 4-7: Query-by-example audio retrieval interface

4.7.2.2. Query-by-keyword audio retrieval

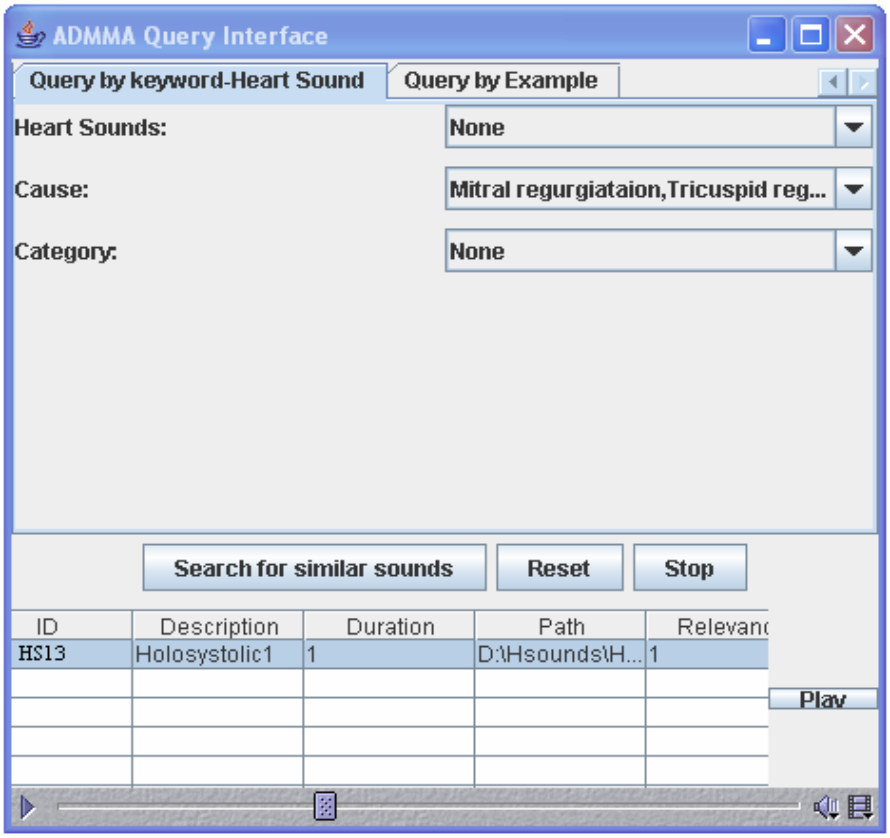
Query-by-keyword audio retrieval provides an interface to facilitate retrieval based on the high-level semantics of audio data. In this interface, users are allowed to give domain dependent keywords that are associated to audio data and listen to returned audio documents. In addition, the query-by-keyword audio retrieval offers random accesses to an audio so as to enable users listen to only part of the audio (for speech) they are interested in. Figure 4-8 depicts the screenshot of the query-by-keyword audio retrieval interface.

Figure 4-8(a) shows the keyword-based heart sound retrieval. Using this interface, users are allowed to select from the list of heart sounds, causes and heart sound categories (normal and abnormal) in their query. It is possible to select one of the above stated options or to use them together. Then, heart sounds that fulfill the given criteria will be listed in the table when the

"search" button is clicked. To listen to these audios, the user can select the record and click on "play selected audio". For instance, in this (Figure 4-8(a)) query the user is interested to list heart sounds that are a result of "Acute Mitral regurgitation" without specifying a particular heart sound and category.

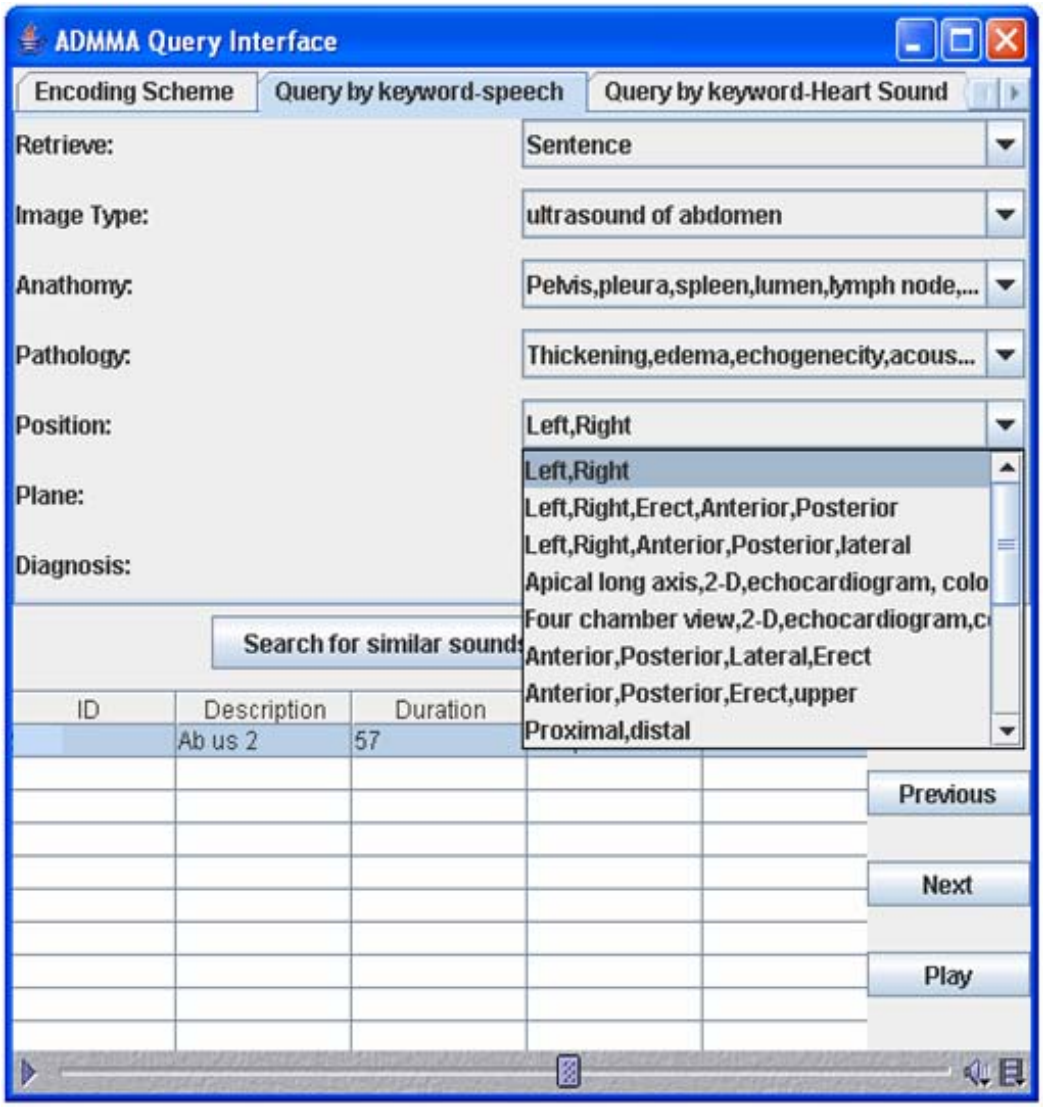
The query can be stated using the traditional SQL language as follows:

Select *HSID, Description, Duration, Path, Relevance*
From *HSound*
Where *(HSound.M.cause="Acute Mitral regurgitation");*



(a) Keyword-based heart sound retrieval

Figure 4-8(b) depicts the keyword-based speech retrieval interface. Here, users are allowed to select from the different values of application dependent keywords. The user can select which speech unit (topic, sentence) he/she is interested in from the "Retrieve" box. After selecting the possible values, the user will click on "search" button. Documents that are assumed to be relevant will be displayed in the table where a user can click to "play" and listen to the audio. It also enables users to listen to the preceding and following sentences by clicking the "Prev" and "Next" buttons respectively.



(b) Keyword-based speech retrieval

Figure 4-8: Screenshot of keyword-based audio retrieval interface (a) and (b)

The query that illustrates Figure 4-8(b) is given as:

Select *SID, Description, Duration, Path, Relevance*
From *Sent*
Where *(Sent.Ms.ImageType="Computer Tomography") And (Sent.Ms.Anatomy="Lung") And (Sent.Ms.Pathlogy="density") And (Sent.Ms.Position="Posterior") And Longitudinal") And (Sent.Ms.Diagnosis="Metastisized Cancer")*

4.8. Summary

In this chapter, the prototype system (ADMMA) that is developed to demonstrate the practicality of the proposed models is presented. To benefit from the Java's platform/database independent functionality and its various integral APIs, we have implemented ADMMA using Java. We used Oracle9i intermedia to manage audio object content in an integrated fashion with the traditional alphanumeric data. We have also presented the general architecture of ADMMA, the different layers that are incorporated in the architecture as well as the representation of ADMMA based on the proposed audio repository model. Lastly, we discussed the similarity matching used in our prototype, which is the Hamming Distance. Using this distance metric, content-based audio retrieval is demonstrated. Furthermore, by extracting domain dependent keywords, we have shown the application of high-level metadata in retrieving audio data. In order to give a general highlight to what functionalities the prototype supports, the different interfaces incorporated in our prototype are presented.

In general, with the aid of ADMMA, the applicability of our proposed models in multi-criteria query formation and retrieval with practical application in the domain of Medical audio data management is demonstrated.

5. Experimental Results

5.1. Preliminary Experiments

This section describes the experimental setup chosen for the preliminary experimentation carried out to select appropriate features for the domain considered in this study.

5.1.1. Data preparation

For audio research in a specific domain, the most desirable option is often to find a data that has been collected previously and is in use by other researchers, as this provides a point to start with and a set of works with which to compare results. Since, such data is unavailable currently, we tried to collect some sounds that we believed are relevant for our study. This subsection presents sample data used in our experimentation. The sample data is grouped into heart sounds and speech recordings.

Heart sounds

There is no publicly available reference database of heart sounds. Thus, a custom database of 48 heart sounds having 6 classes (each having 8 sounds) is built from an Internet search, medical learning CDs and live recording. The file lengths, compression type, sampling rates, and background noise levels vary over the samples. Thus, the first step that we have taken is bringing those signals in to a common representation or a uniform standard format - compression type, sampling rate, quantization rate and number of channels. This step is considered to be the pre-processing stage of the feature extraction by which the audio signal is converted into a fixed target format with predefined settings. This is followed by the actual feature extraction. In the present configuration, an off-the-shelf audio editing tool, Audacity, is used to make fine-tuning on the sample audio collection. With the help of Audacity, background noises and other additional sounds are eliminated as well as audio quality is kept constant. All sounds are kept uniform in terms of nearly equal duration, sampling rate of 44100Hz and quantization rate of 16 bits. In addition, these audios are made to be single channel and of the same compression type (WAV). The need for such pre-processing step is important in order to control the effect of such encoding variations on feature extraction and retrieval tasks. The chosen sampling and

quantization rate are chosen because the standard highest frequency for most computers is 44.1KHz and in terms of bit rate, 16 bits per sample is considered good enough. Furthermore, the feature extraction tool used in this work is tailored to perform in such representations.

Speech recordings

The second group of sample data is collected from live recording of physicians while describing radiological and echo-cardiological images. 30 speech recordings are collected in this way considering different medical cases of patients. Among the different factors that affect speech signals are room acoustics, microphone characteristics and background noises. In addition, spoken language is characterized by disffluent phenomena such as incomplete sentences, hesitations and repetitions, which result in faulty and complicated analysis [55]. These problems have also been reflected in our collection. By means of the previously stated audio editing tool ("Audacity"), we have tried to get rid of unwanted background noises, repetitions and hesitations. Furthermore, some of the preprocessing made to heart sounds is also applied to speech for the sake of uniformity.

5.1.2. Feature extraction

The goal of feature extraction is to describe the perceptual properties of a signal using small number of parameters [67]. After completion of data preparation and preprocessing tasks, the next logical step is extraction of features that are believed to be suitable for the application under consideration. The effectiveness of an audio feature is reliant in the domain to which it is used. Features that are perfectly used in the speech community might not perform well in that of music or vice versa. To alleviate such problems, the MPEG-7 audio standard has provided a large set of low-level audio descriptors [68]. These descriptors are part of many state-of-the-art audio retrieval systems [69,70]. This standardized descriptor design forms the basis for achieving an open platform for automatic audio identification. MPEG-7 LLDs are generic signal characteristics, useful for many purposes and applicable to a wide range of signals. Although MFCCS are good choices for general audio retrieval, as indicated in section 2.1.4, different investigations have shown that MPEG-7 descriptors perform comparably to MFCCs and even sometimes outperform them [70]. Consequently, this work bases itself and tries to exploit low-

level signal features standardized within the MPEG-7 framework. The MPEG-7 LLD extractor that is provided by the Technical University of Berlin (TUB) is used to extract MPEG-7 LLDs from input audio signals [71].

Selecting appropriate (relevant to the application) audio data descriptors from MPEG-7 descriptors is another subject worth looking. In the area of audio signal description, MPEG-7 audio provides generic frameworks that are pertinent for diverse application domains. As these descriptors are considered to form a universal toolbox for many applications, all of them are not expected to be compulsory in regard to a specific audio class. Thus, the preliminary experiment stated in this subsection tries to identify those features that best suite for signals like heart sounds. Extracting low-level features for speech signals is beyond the scope of this study and all feature extraction issues are applied to only heart sounds. Among the MPEG-7 LLDs, we have chosen some features based on their ability of robust identification for generic audio classes. These descriptors were chosen from a range of competing features as they were found to exhibit good performance in a variety of applications in audio identification. In the following, we will explain some of the descriptors used in our preliminary experiment.

AudioSpectrumFlatness, as described in section 2.1.4, is one of the MPEG-7 descriptors that define procedures for extracting relevant feature information from an audio signal. It is selected as it has been shown to be robust with respect to a wide range of distortions [72] and can be represented in a very compact fashion [73]. This element is part of the general MPEG-7 Audio "toolbox" of LLDs and is destined to be universal in its application.

Harmonic features are also set of descriptors that worth exploiting as they characterize the harmonic structure of a signal. Harmonic features are opted to be contending candidate features as they are good in representing periodic or quasi-periodic signals where heart sounds belong to. HarmonicSpectralCentroid, HarmonicSpectralDeviationType, HarmonicSpectralSpread, HarmonicSpectralVariation and AudioHarmonicity are different types of harmonic features that are to be tested for representing heart sounds.

AudioSpectrumBasis and AudioSpectrumProjection are statistical basis functions of a spectrogram that are used for dimension reduction and summarization of the spectrogram. These features are considered here as they can be utilized in general-purpose sound recognition.

Other MPEG-7 features are left intentionally as they are especially destined to be useful for particular classes of sounds. For instance, **TimbralTemporal** and **TimbralSpectral** features are valuable for the description of musical timbre and used typically in music analysis and their low power in expressing other sounds is tested in [74].

5.1.2.1. Test setup description

The sound collections used in this experiment are small to draw general conclusion, however, they are still in a range one would call reasonable. We extract all the aforesaid features from the 48 heart sounds, form a feature vector, and store the vector in a database together with the source audio. To do this, we followed the following procedures. In the first step, MPEG-7 LLDs are extracted from the audio samples with the LLD extractor provided by TUB. In the second step, the feature vectors of each audio sample are read in to an *M*x*N* sized matrix, where *M* is the number of frames formed and *N* is the number of frequency bins or vector sizes. The third step taken is dimension reduction and summarization via statistical tools (i.e. frame-based features are summarized by their means and variances in order to obtain descriptions of entire audio frames). The resulting multidimensional feature vector is, then, used as the basis for similarity matching between the audio signals. Lastly, by giving one heart sound at a time as an example audio, we computed Euclidean Distance (see equation 1) between the feature vector of the example input and the ones in the database to measure their similarity.

Euclidean distance (d) between two vectors $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$, in Euclidean n-space, is defined as [12]:

$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad \ldots\ldots \qquad [1]$$

We evaluated the quality of features for the application domain empirically in terms of the precision and recall of the top-ranked matches returned to the user, where the baseline assessment is based on personal/subjective domain expert evaluation. Recall is the proportion of retrieved relevant documents of all relevant documents in the database while precision is the percentage of relevant documents retrieved in relation to the total number of documents retrieved [75].

Let *Ret* be the set of retrieved documents and *Rel* the set of relevant documents in the database. Where *No* refers to number of,

Recall R is defined as:

$$\mathbf{R = No\ (Ret \cap Rel)/No\ (Rel)} \quad \ldots\ldots \quad [2]$$

Precision P is defined as:

$$\mathbf{P = No\ (Ret \cap Rel)/No\ (Ret)} \quad \ldots\ldots \quad [3]$$

In the evaluation procedure, one sample at a time was drawn from the database to serve as an example query and the rest were considered as samples in a database. A database sample is considered correctly retrieved, when the sample falls in the same class as that of the example sound. The query is repeated taking each of the 48 samples as an example audio resulting in 2256 pairwise comparisons. The number of correctly retrieved samples in each class *u* (C_u) is calculated for each class $u \in \{1,2,3,4,5,6\}$ by considering the *top ten* returned results. The average of recall of each query is given by:

Recall (u) = $\frac{C_u}{S_u(S_u - 1)}$, where S_u is the total number of samples in the class u.

The average of precision of each query is given by:

Precision (u) = $\frac{C_u}{R_u}$, where R_u is the total samples retrieved for class *u* examples

5.2. Results

In this section, the results of the preliminary experiments are presented. High Recall values indicate high identification rates. High Precision values suggest that only a small number of documents not similar to a given example audio are retrieved. The following tables (Table 5-1 to Table 5-10) address results obtained with the previously discussed audio features.

5.2.1. Combined feature

The first experiment is made by combining all audio features listed in section 5.1.2. These features are used to identify each heart sounds listed in Table 5-1. Table 5-1 shows the Recall and Precision computed using the combined features.

Table 5-1: Combined feature evaluation

Heart sound name	Recall	Precision
Gallop	12.5%	8.8%
Holosystolic murmur	14.3%	10%
Late Systolic murmur	12.5%	8.8%
Mid Systolic murmur	0.0%	0.0%
Normal Heart Sound	32.1%	22.5%
Systolic ejection murmur	0.0%	0.0%
Average	11.90%	8.35%

As indicated in the table, the aggregate features do not suit well for the given data as the average results are 11.9 and 8.35, which are very low. For more general statements about the feature combination, more test data are needed. However, to see the impact of individual features we have conducted the experiment taking each feature separately.

5.2.2. Individual features

As mentioned above, the combined features don't give promising results in identifying an example audio data. Therefore, the experiments are conducted using each feature separately to evaluate its discriminative power. The following are the list of applied audio features together with their results.

AudioSpectrumFlatness(ASF)

AudioSpectrumFlatness is one of the candidate audio low-level features that are used in audio identification. As indicated in the previous subsections, first, the AudioSpectrumFlatness of each heart sound is extracted and stored in the database. Then similarity matching is made based on the distances computed between the AudioSpectrumFlatness values. All the subsequent experiments are made with the same manner.

As indicated in Table 5-2, the AudioSpectrumFlatness, yields predominantly good results. These results are consistent throughout all heart sound classes as shown by relatively high recall/precision pairs attained.

Table 5-2: Evaluation of "AudioSpectrumFlatness"

Heart sound name	Recall	Precision
Gallop	80.0%	56.3%
Holosystolic murmur	78.6%	55%
Late Systolic murmur	82.1%	57.5%
Mid Systolic murmur	71.4%	50.0%
Normal Heart Sound	91.1%	63.8%
Systolic ejection murmur	69.6%	48.8%
Average	78.8%	55.2%

HarmonicSpectralCentroid(HSC)

HarmonicSpectralCentroid performs poorly as it is indicated in table 5-3. Only the "Late Systolic murmur" heart sound subclass is better identified when compared to other subclasses. The general assessment is that this feature failed to distinguish between the different audio classes as shown in the low results of recall and precision.

Table 5-3: Evaluation of "HarmonicSpectralCentroid"

Heart sound name	Recall	Precision
Gallop	26.8%	18.8%
Holosystolic murmur	28.6%	20.0%
Late Systolic murmur	39.3%	27.5%
Mid Systolic murmur	19.6%	13.8%
Normal Heart Sound	28.6%	20.0%
Systolic ejection murmur	16.1%	11.2%
Average	26.5%	18.6%

HarmonicSpectralDeviation(HSD)

HarmonicSpectralDeviation performs in a similar fashion to HarmonicSpectralCentroid. The results are poor for almost all sound classes as indicated in table 5-4. These results at least show that HarmonicSpectralDeviation is not discriminative to be used as a reliable feature for the data under consideration with the given sample size.

Table 5-4: Evaluation of "HarmonicSpectralDeviation"

Heart sound name	Recall	Precision
Gallop	23.2%	16.3%
Holosystolic murmur	25.0%	17.5%
Late Systolic murmur	17.9%	12.5%
Mid Systolic murmur	35.7%	25.0%
Normal Heart Sound	32.1%	22.5%
Systolic ejection murmur	26.8%	18.8%
Average	26.8%	18.8%

HarmonicSpectralSpread(HSS)

HarmonicSpectralSpread is not able to identify the different heart sounds as it is indicated in table 5-5. Except for "Mid Systolic murmur" and "Normal Heart Sound", which are identified better compared with the others. As a general evaluation, HarmonicSpectralSpread has hardly any discriminative power.

Table 5-5: Evaluation of "HarmonicSpectralSpread"

Heart sound name	Recall	Precision
Gallop	19.6%	13.8%
Holosystolic murmur	17.9%	12.5%
Late Systolic murmur	17.9%	12.5%
Mid Systolic murmur	33.9%	23.8%
Normal Heart Sound	35.7%	25.0%
Systolic ejection murmur	21.4%	15.0%
Average	24.4%	17.1%

HarmonicSpectralVariation(HSV)

HarmonicSpectralVariation yields slightly better results compared to the previously discussed features. The results obtained by this feature are shown in table 5-6. In contrast to other classes of sounds, "Normal heart sound" is relatively identified well. However, as a general evaluation, HarmonicSpectralVariation doesn't have enough discriminative power that would make it useful for identification of sounds for the audio domain considered.

Table 5-6: Evaluation of "HarmonicSpectralVariation"

Heart sound name	Recall	Precision
Gallop	23.2%	16.3%
Holosystolic murmur	32.1%	22.5%
Late Systolic murmur	26.8%	18.8%
Mid Systolic murmur	28.6%	20%
Normal Heart Sound	42.9%	30.0%
Systolic ejection murmur	25.0%	17.5%
Average	29.8%	20.9%

AudioSpectrumBasis(ASB)

AudioSpectrumBasis exhibits slightly better identification power than all the previously discussed features except AudioSpectrumFlatness. The obtained results are presented in table 5-7. As indicated in the table, most of the classes are better identified relatively. However, these results are far from satisfactory for practical use and shows that AudioSpectrumBasis is limited in its expressiveness.

Table 5-7: Evaluation of "AudioSpectrumBasis"

Heart sound name	Recall	Precision
Gallop	30.4%	21.3%
Holosystolic murmur	26.8%	18.8%
Late Systolic murmur	35.7%	25.0%
Mid Systolic murmur	23.2%	16.3%
Normal Heart Sound	41.1%	28.8%
Systolic ejection murmur	32.1%	22.5%
Average	31.6%	22.1%

AudioSpectrumProjection(ASP)

Performance of AudioSpectrumProjection is slightly above the performance of HarmonicSpectralDeviation and HarmonicSpectralCentroid. As it is illustrated in table 5-8, AudioSpectrumProjection doesn't carry enough discriminative information for identifying the audio data considered in this experiment.

Table 5-8: Evaluation of "AudioSpectrumProjection"

Heart sound name	Recall	Precision
Gallop	21.4%	15.0%
Holosystolic murmur	32.1%	22.5%
Late Systolic murmur	26.8%	18.8%
Mid Systolic murmur	23.2%	16.3%
Normal Heart Sound	33.9%	23.8%
Systolic ejection murmur	32.1%	22.5%
Average	28.3%	19.8%

AudioHarmonicity(AH)

AudioHarmonicity yields results on a higher level than identification with all other features next to AudioSpactrumFlattness. "Late Systolic murmur" and "Normal Heart Sound" are found to be well identified compared to other classes. The results in table 5-9 indicate that AudioHarmonicity may be used in combination with other better performing features, but do not contain enough discriminative information that would make them useful for classification as a single discriminative feature.

Table 5-9: Evaluation of "AudioHarmonicity"

Heart sound name	Recall	Precision
Gallop	51.8%	36.3%
Holosystolic murmur	48.2%	33.8%
Late Systolic murmur	60.7%	42.5%
Mid Systolic murmur	53.6%	37.5%
Normal Heart Sound	69.6%	48.8%
Systolic ejection murmur	55.4%	38.8%
Average	56.6%	39.6%

As indicated in the tables above, AudioSpectrumFlatness exhibits a good retrieval and discrimination power followed by AudioHarmonicity. Most of the heart sounds in the database come from the abnormal heart sound (those sounds other than normal) category. Such an abnormality reduces the rhythmic and periodic characteristics of those sounds. This can be one of

the probable logical explanations that can be given to most of the harmonic features for failing to discriminate the different heart sounds.

AudioSpectrumFlatness and AudioHarmonicity

As the good performing features, AudioSpectrumFlatness and AudioHarmonicity form the basis of feature combinations discussed.

Table 5-10 illustrates the results of their combined feature in identification of a given audio.

Table 5-10: Combined feature of AudioSpectrumFlatness and AudioHarmonicity

Heart sound name	Recall	Precision
Gallop	32.1%	22.5%
Holosystolic murmur	41.1%	28.8%
Late Systolic murmur	32.1%	22.5%
Mid Systolic murmur	35.7%	25.0%
Normal Heart Sound	23.2%	16.3%
Systolic ejection murmur	30.4%	21.3%
Average	32.4%	22.7%

Even though individual features of AudioSpectrumFlatness and AudioHarmoncity show good results their combined feature doesn't show the expected identification power. Thus, only AudioSpectrumFlatness will be used for further analysis.

In Figure 5-1 and Figure 5-2, the computed results of recall and precision of the experiments are presented for easier understanding. As it can be seen from the chart, AudioSpectrumFlatness outperformed the individual and combined features in both recall and precision values followed by AudioHarmonicity. Moreover, the combined features of AudioSpectrumFlatness and AudioHarmonicity have achieved the third result next to AudioHarmonicity. On the other hand, AudioSpectrumProjection and HarmonicSpectralCentroid have shown the least results in both recall and precision values.

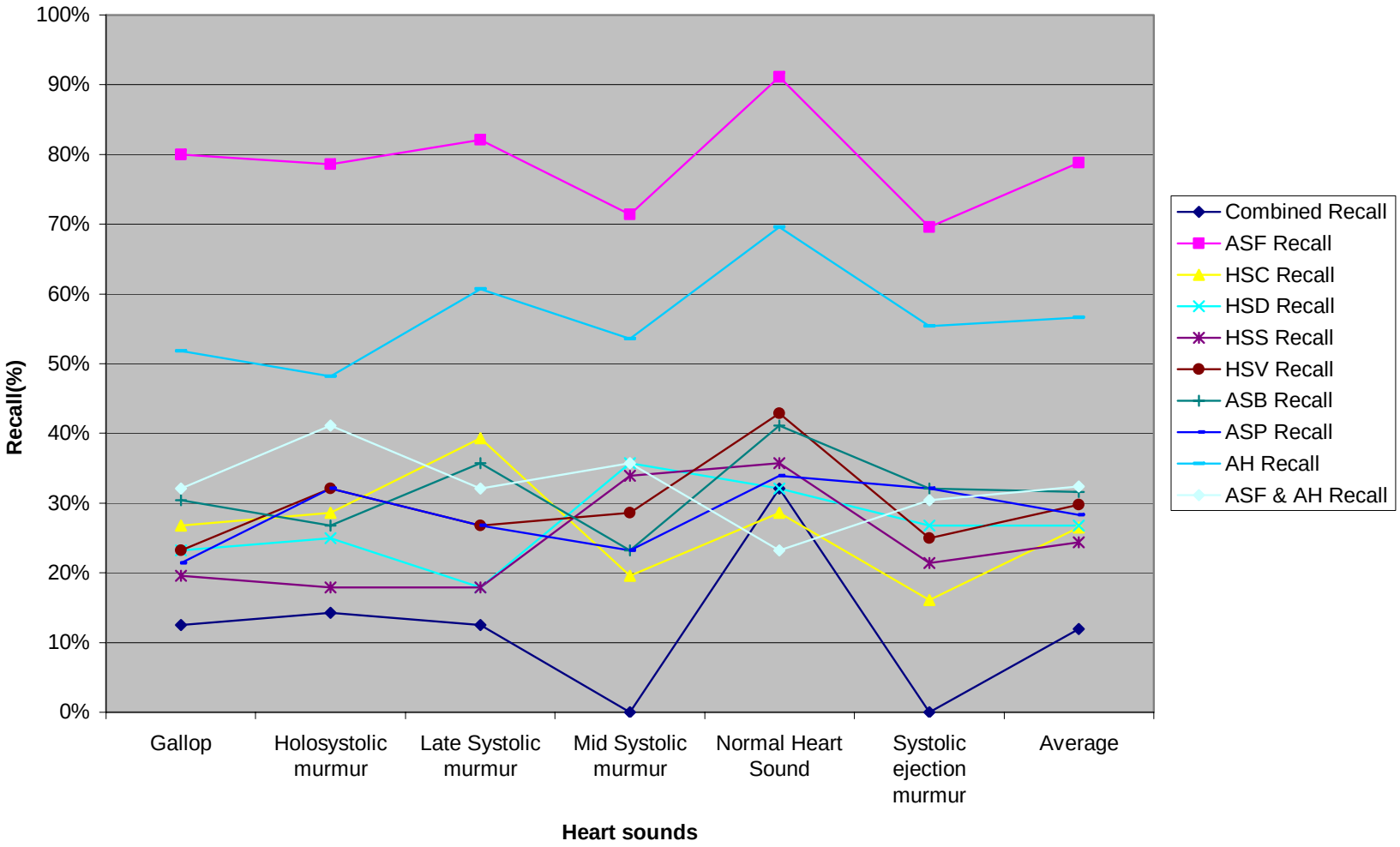


Figure 5-1: Recall results of features considered in the evaluation

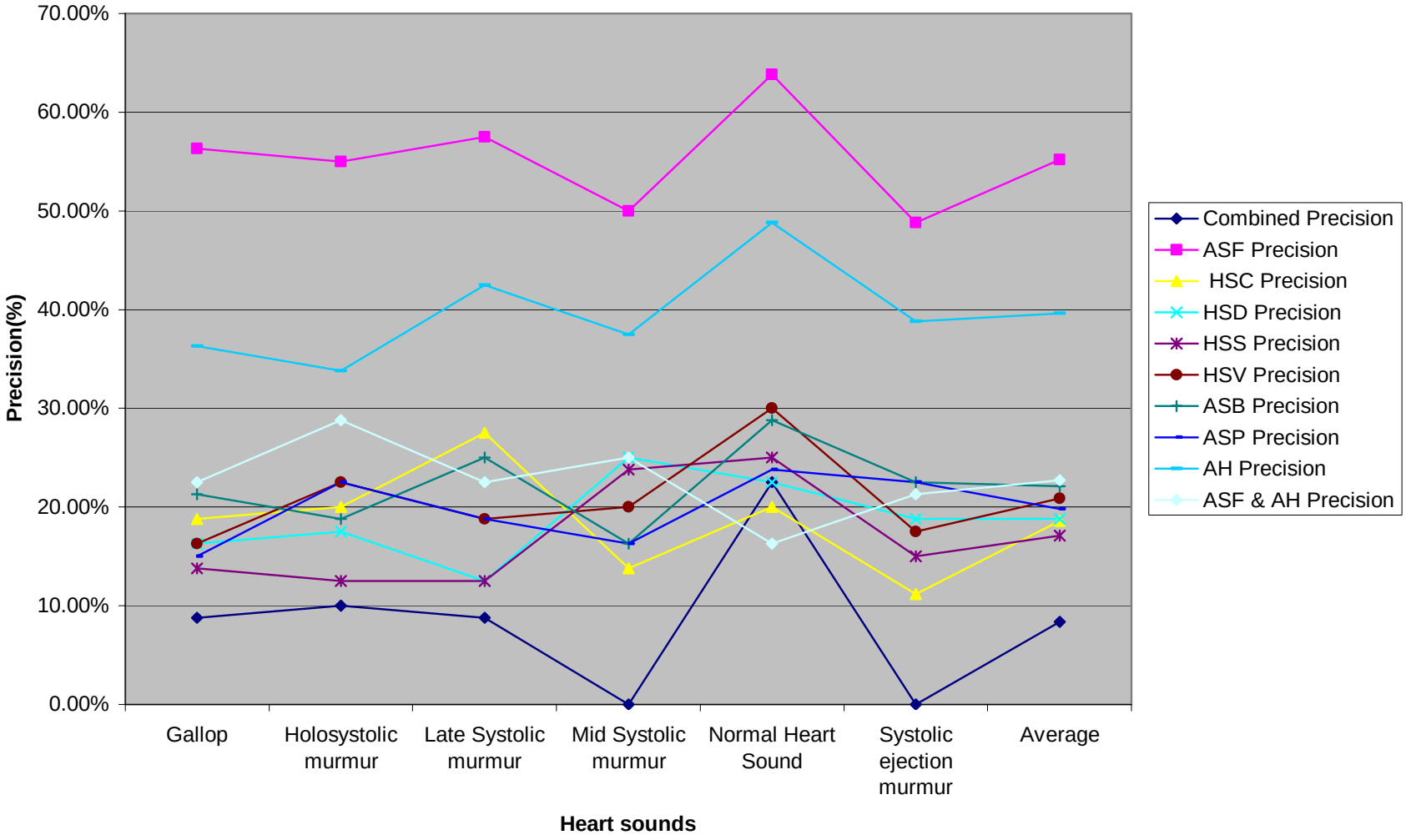


Figure 5-2: Precision results of features considered in the evaluation

Figure 5-3, summarizes the overall processes accomplished in the preliminary experiments in order to select appropriate low-level audio features for the specific domain considered in our prototype.

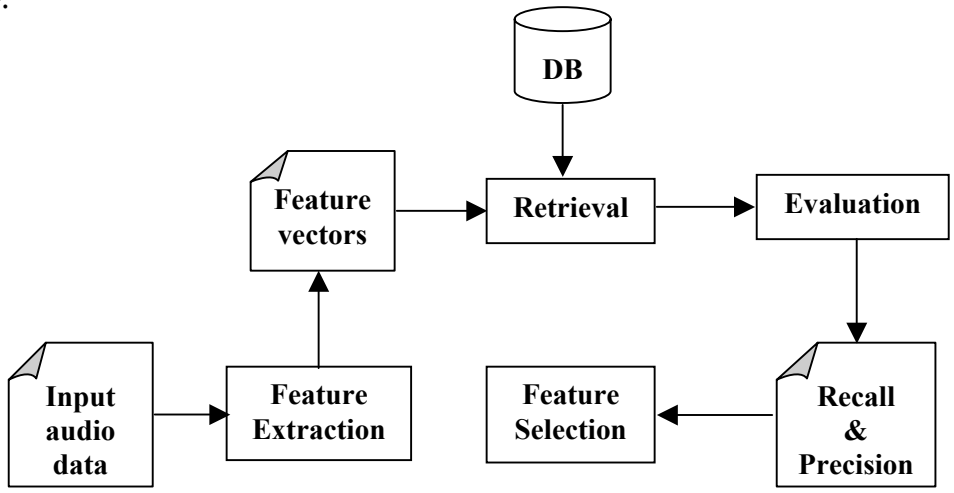


Figure 5-3: Experimental process for feature selection in content-based audio retrieval

5.3. Experiment using fingerprints

An audio fingerprint, sometimes known as a robust hush, enables an unknown audio to be identified by comparing its fingerprint to the fingerprints contained in the database.

The basic idea of audio fingerprinting is to identify a piece of audio content by extracting unique signature from it. As it is discussed in section 2.1.4, AudioSpectrumFlatness is intended to perform such fingerprinting. Thus, motivated by the results of AudioSpectrumFlatness, fingerprints are extracted and used as low-level features for representing heart sounds in content-based audio retrieval.

The fingerprint extraction scheme employed in this thesis is based on the following general approach. The audio signal is first segmented into *overlapping* frames. The overlapping frames have a length of 0.37 seconds. This results in the extraction of a 32-bit sub-fingerprint for every 11.6 milliseconds. A sub-fingerprint is a compact representation of a single frame. A single sub-fingerprint usually doesn't contain sufficient data to identify an audio signal. The basic unit that contains sufficient data to identify an audio will be referred to as a *fingerprint block*. A fingerprint block consists of 256 subsequent sub-fingerprints. The output of this algorithm is a matrix having a dimension of (Number Of Frames X32).

Finally, the fingerprints of audio signals are compared using hamming distance, which measures the number of disagreements between two vectors. Two fingerprint blocks are declared similar if their Hamming distance, usually expressed as Bit Error Rate (BER), is below certain threshold. In [67], it is shown that a BER of less than 35% leads to a very reliable identification, the same threshold have been applied in our experiment. This means that out of 8192(32x256) bits there must be less than 2867 bits in error in order to decide that the fingerprint blocks are similar. The corresponding results are illustrated in table 5-11.

Table 5-11: Results based on audio fingerprints

Heart sound name	Recall	Precision
Gallop	78.6%	55.0%
Holosystolic murmur	83.9%	58.8%
Late Systolic murmur	85.7%	60.0%
Mid Systolic murmur	82.1%	57.5%
Normal Heart Sound	87.5%	61.3%
Systolic ejection murmur	80.4%	56.3%
Average	83.0%	58.2%

5.4. Results of the experiment using fingerprints

Meaningful evaluation is crucial for any retrieval system to demonstrate the importance of chosen features and methods. The general results attained using audio fingerprints are presented in the following subsections.

5.4.1. Content-based audio retrieval

The audio retrieval process that employs the proposed models and the low-level features (fingerprints) discussed herein function efficiently and return correct results the majority of the time. Good results have been attained through fingerprints as indicated in table 5-11. Consistent results are achieved almost for all classes except for “Gallop”, which is relatively less identified. Though it is not possible to have a general conclusion due to small number of classes and number of samples, the overall assessment is regarded to be satisfactory. Further more, what has been

seen during the experiment gives a good direction in selecting features for signals like heart sound for a larger dataset.

5.4.2. Keyword-based audio retrieval

The above-mentioned evaluations are based on results of content-based audio retrieval in which input query is given as an example audio. In the upcoming subsection, we will see audio retrieval based on keywords that are used to retrieve speech data. As it is stated at the beginning of the chapter, speech recordings of image description are captured and structured into constituent units. The first step in structuring speech is to identify keywords that are amenable to a given speech. To do so, we involved domain experts to select representative keywords for each medical image description considered in this study.

We tried to retrieve speech data based on the semantic content they encompass rather than the acoustic characteristics they reveal. The relevance of query responses is evaluated by making a simple usability study of our prototype by experts in the medical application domain. Six physicians (users) who are experienced Internet users (familiar to search engines) and who were not consulted while selecting representative keywords (new for the system) are asked to provide queries to retrieve image descriptions and 15 such queries are collected. The physicians were given no instruction on the use of our system.

The goals of this usability study are:

- To know what information is expected to be discovered by the system and to ensure that selected keywords for the given images are generic so as to be used by any expert in the domain.
- To ensure that query responses are as per the expectation of experts in the domain.

Users have tried to retrieve part of the speech document (sentence, topic/speech) as per the structure proposed in our repository model. One observation is that majority of the users - 4 of them (66.7%) were interested to listen the whole speech whereas 2 of them (33.3%) where inclined to retrieve speech at sentence level. Among the 15 queries applied for the speech based image description, 12 of them (80%) are supported by the system. The reason of failing to

respond for the 20% is that some of the keywords were new for the system (i.e. they were not included in the keyword list). All 12 queries are responded correctly (100% precision/recall pairs have been attained). However, from our results, one can understand universally usable keywords are compulsory and a standard keyword usage needs to be developed in order to benefit from keyword-based audio retrieval systems in the medical application domain.

In addition, keyword-based retrieval has been done in identifying heart sounds. Two classes of keywords are used. On one hand, retrieval is made based on descriptions associated to heart sounds, which give 100% precision/recall pairs. On the other hand, retrieval is made by providing the system with possible causes for an abnormal heart sound, which again results 100% precision/recall pairs.

5.5. Summary

In this chapter, the experiments conducted and their results are discussed. The experiments are divided in to two. The first is the preliminary experimentation, which is used to select appropriate low-level features that best suite for the audio type involved in this study. The second is the experiment that is used for evaluating the system with respect to the proposed models. The preliminary experimentation does not involve the speech part of our prototype since dealing with low-level features of speech documents are beyond the scope of this study. The feature selection experiments are split into two test series. In the first run, all features are combined to form a unit multidimensional feature vector. In the second run, selected features are tested individually. The combined and most of the individual features failed to be used for audio identification for the sample considered. On the other hand, encouraging results have been obtained with AudioSpectrumFlatness in detecting similar sounds. Consequently, audio fingerprint is used to identify unknown heart sounds and has shown good representative and identification power in our prototype.

The second group of our experiment is concerned in keyword-based audio retrieval. Both heart sounds and speech based image descriptions are tested. The queries provided for both groups attained good results, which show the applicability of our models in supporting audio retrieval by using their high-level content. The results achieved during speech retrieval have shown that structuring speech into its constituent units enables accessing large speech archives randomly.

6. Conclusions and Future works

In the first and second chapters of this thesis, we have tried to show the applications and basic challenges that are associated with audio data management. As it is known, the amount of available audio data is in rapid increase, and retrieving audio information is becoming more and more difficult. However, despite this increased volume and interest, modeling and representation and retrieval of audio data have received relatively little attention. This is due to the unique challenges that audio data poses. Audio is a content rich media with complex structure. One of the complexity stems from the unstructured and heterogeneous behavior of generic audio. A general audio might contain different classes that exhibit different acoustic and perceptual characteristics. Thus, the first step in any audio analysis is to classify audio into classes that exhibit homogenous characteristics and then apply specific techniques that suit to each of these classes. Though audio classification issues are beyond the scope of this thesis, we have categorized generic audio coarsely into speech, music and environmental sounds in our proposed models. We have also discussed that once this categorization is done, the different audio types can be further analyzed by more appropriate techniques such as speech recognition, music information retrieval etc.

The increase in the amount of audio data deployed and used in today's applications not only caused audio to draw more attention as a multimedia data type, but also led to the requirement of efficient management of audio data. One such requirement for audio management is proper modeling of audio data. To provide efficient search and retrieval of audio data from large archives, we need to model audio data appropriately. In this thesis, an audio data model and audio data repository model are proposed. Furthermore, techniques for multi-criteria query formation are presented. Finally, the models and techniques proposed are evaluated for audio data management requirement in a medical application domain. The purpose of modeling audio data is to identify features that must be captured to facilitate queries and operations that are to be performed on audio data. In order to develop effective tools for interacting with audio collections, it is important to model high-level information together with the low-level ones. Thus, we identified the low-level and high-level features that should be captured to represent an audio data

and proposed a data model that incorporates both metadata (high-level features) and low-level features.

In addition, such audio data models are used to structure the data to reflect the inherent relationships that exist between the various items of audio data. The proposed audio data model does not treat all audio data as an indivisible data objects. For instance, speech is structured to its units in order to provide users with only a segment of speech in which they are interested in. Structuring speech data enables users to have a random access of an audio segment of particular interest instead of auditing the entire audio from start to end, which takes significant time.

Conventional database systems are designed for managing textual and numerical data, and retrieving such data is often based on simple comparisons of text/numerical values. However, this simple method of storage and retrieval is no longer adequate for audio data, since the digitized representation of audio does not convey the reality of this media item. Therefore, in addition to the audio data model, an audio repository model that enables audio data to be stored in DBMSs in a convenient way by taking the intrinsic features of audio data into account is proposed. This repository model is proposed from the point that database systems managing audio information should provide support for a diverse range of applications. The proposed repository model is not based on a single importance, but is closely related to an underlying data model and the scheme for representing queries because the core of an audio retrieval system is the database that stores the audio document.

The proposed repository model stores annotated metadata and extracted features. Low-level features are automatically computed by a feature extraction mechanism. However, low-level audio description cannot provide a semantic interaction with audio contents. The major problem is the apparent gap between low-level audio features and high-level semantics because similarities in low-level features do not always match user perceptions and lacks semantic interpretation. The semantic gap refers to the mismatch between high-level concepts and low-level descriptions and this semantic gap is positioned between the content of media and textual information describing the semantics of the content. Thus, modeling high-level information with low-level information is indisputable.

The most common approach to capture high-level information has been to manually annotate audio signals with additional information. Modern systems support automatic extraction of annotations, as in the case of ASR. However, it is not possible to generate a verbose and detailed description in unconstrained domains because ASR systems suffer from Out Of Vocabulary (OOV) problems. Additionally, algorithms that automatically extract description from audio files are generally not mature enough to provide the user with friendly representation that users demand when interacting with audio content. Thus, though time-consuming and error-prone, human labeling is a common practice. In our practical demonstration, domain specific keywords are selected and stored as high-level information manually.

Our proposed retrieval system supports multiple types of queries. Once the low-level and high-level descriptions are stored, a search and retrieval module queries the database to get required audio segments. Query-by-example techniques directly use documents as query objects. The retrieval system computes features from the query documents and tries to locate similar documents in the database by applying a similarity measure. The other method used in our thesis is query-by-keyword, where the user defines the desired class of documents or terms describing the documents. Query-by- keyword makes use of media annotations stored in the database. With the proposed model, a multi-criteria query that considers the different features (content and semantics) of audio can be formulated for a wide area of applications.

Lastly, as mentioned previously, our tasks involve developing an audio data model and audio repository model that combines both the low-level and high-level features of an audio signal so as to enable audio and text-based retrieval of audio document. Therefore, using a practical demonstration, we have shown that systems that combine audio retrieval based on structured text descriptions (metadata) and content-based audio retrieval techniques may offer the best way forward, meeting the goals of the study, as set out in section 1.3. We hope that this paper will stimulate some research in the field, which still needs to be largely explored, as there are still many works that can be done in this regard.

Major contributions of our study are given as follows:

- The basic components of audio data that needs to be captured for representing and managing audio data are identified,

- An audio data model that discerns the structure of an audio document together with their-low-level and high-level information is designed.
- High-level information that needs to be captured to an audio document are identified, classified and modeled together with the low-level information
- An audio data repository model that is suitable to be used in the management of audio data under the context of an Object Relational (OR) paradigm is proposed.
- Appropriate low-level features that are suitable for the application domain considered are identified and a prototype named ADMMA is developed to demonstrate the applicability of the proposed models and the multi-criteria query formation in the area of medical audio data management.

Future works

- The key to any QBE system is in the definition of audio similarity. Thus, audio recognition engines must be developed, as there is no standard defined for similarity operations in the audio domain.
- Many of application scenarios for audio retrieval rely on the flexible use of audio retrieval in arbitrary environments, which leads to the need of audio retrieval systems that run on small handheld devices such as Personal Digital Assistant (PDAs).
- Short summary of a sound file that captures essential elements of the original sound file is needed to enable users to go through results easily.
- Another interesting application is to apply audio retrieval in a multimodal environment, where visual, textual, and acoustic information can be combined to take advantage of synergetic effects.
- Describing the whole audio data is a very time consuming process, thus, automatic generation of mappings between low-level features and high-level semantics is highly desirable.
- Keyword-based audio retrieval approaches can be made possible. But, due to the limitations in natural languages, designing thesaurus for the keywords included in the metadata information is still a problem that needs to be solved.

REFERENCES

[1] E. Weis, Sync tanks: The art and technique of postproduction sound. Cineaste,1995, 21(1):56.

[2] G. Tzanetakis and P. Cook. Manipulation, Analysis and Retrieval Systems for Audio Signals. PhD thesis, Princeton University, 1999:

[3] A. Flexer, E. Pampalk, and G. Widmer. Novelty detection for spectral similarity of songs. ISMIR, 2005.

[4] R. B. Dannenberg and N. Hu. Understanding search performance in query-by-humming systems. ISMIR, 2004.

[5] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming: Musical information retrieval in an audio database. In The Third ACM International Multimedia Conference and Exhibition (MULTIMEDIA ’95),1995,pages 231–236.

[6] M. Slaney. Semantic-audio retrieval. In Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002, volume 4, pages IV4108–11.

[7] A.I. Zayed. Advances in Shannon’s Sampling. Theory, CRC Press, Boca Raton,1993, pp.157-159.

[8] G. Tzanetakis, P. Cook. Multifeature audio segmentation for browsing and annotation. In Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York, Oct 1999. 17-20.,

[9] David F. Walnut, An Introduction to Wavelet Analysis, Birkauser Boston Second, 2002.

[10] J. Saunders. Real-time discrimination of broadcast speech/music. In International Conference on Acoustics, Speech and Signal Processing,1996, pages 993–996. IEEE.

[11] T. Zhang and C. J. Kuo. Hierarchical system for content-based audio classification and retrieval. In Proc. International Conference on Acoustic, Speech, Signal Processing,1998, volume 6, pages 3001–3004.

[12] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. In IEEE Multimedia, 1996, vol. 3, pages 27-36.

[13] G.V. Ramana Rao and J. Srichand. Word boundary detection using pitch variations. In Fourth International Conference on Spoken Language Processing,1996,volume 2, pages 813–816.

[14] Y. Wang, Z. Liu, J.c. Huang. Multimedia content analysis using both audio and visual cues. IEEE Signal Process. Mag, 2000, 17(6), pp. 12-36.

[15] D. Pye. Content-based methods for the management of digital music. In Proc. the international Conference on Acoustics, Speech, and Signal Processing, 2000.

[16] J. Foote. Content-based retrieval of music and audio. In Proc. SPIE,1997, pages 138–147.

[17] M. Liu and C.Wan. A study on content-based classification and retrieval of audio database. In Proc. 2001 International Database Engineering and Applications Symposium,2001, pages 339–45.

[18] D. Mitrovic, M. Zeppelzauer and C. Breiteneder. Discrimination and Retrieval of Animal Sounds, In Proceedings of the IEEE conference on Multimedia Modeling, 2006.

[19] ISO/IEC JTC1/SC29/WG11 (MPEG). Multimedia content description interface - part 4: Audio International Standard 15938-4, ISO/IEC, 2001.

[20] H-G. Kim, N. Moreau, & T. Sikora. MPEG-7 audio and beyond. West Sussex: Wiley,2005.

[21] E. Scheirer and M. Slaney. Construction and evaluation of a robust multi-feature speech/music discriminator. In Proc. ICASSP, Munich,Germany, 1997, pages 1331–1334.

[22] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. Pattern Recognition Letters,April 2001,22:533–544.

[23] E. Singer, M. A. Kohler, P. A. Torres-carrasquillo, and R. J. Greene. Approaches to language identification using Gaussian mixture models. ICASSP, 2002.

[24] B. Whitman, D. Roy, and B. Vercoe. Learning word meanings and descriptive parameter spaces from music. In HLT-NAACL03 workshop, 2003.

[25] S. Esmaili, S. Krishnan, and K. Raahemifar. Content-based audio classification and retrieval using joint time-frequency analysis. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2004,5(17-21):665–668.

[26] P. Koppenberger, M., Groux, S. L., Ricard, J., Herrera, P., and Wack, N. Nearest-neighbor generic sound classification with a wordnet-based taxonomy. In Proc.116th AES Convention, Berlin, Germany, 2004.

[27] Stewart Fist. MPEG in a round hole, August 1998. Available at: http://www.abc.net.au/http/sfist/mpeg1.htm (Consulted on July. 21,2006)

[28] P. Noll. MPEG digital audio coding. IEEE Signal Processing Magazine, September 1997, pages59–81.

[29] I. JTC1/SC29. Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s-IS 11172 (Part 3, Audio), 1992.

[30] I. JTC1/SC29. Information Technology-Generic Coding of Moving Pictures and Associated Audio Information-IS 13818 (Part 3, Audio),1994.

[31] G. Tzanetakis and P. Cook. Marsyas3D: a prototype audio browser-editor using a large scale immersive visual and audio display. In Proc. Int. Conf. on Auditoy Display (ICAD), Espoo, Finland, Aug. 2001.

[32] S-F. Chang, T. Sikora, and A. Puri." Overview of the MPEG-7 Standard". IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on MPEG-7, June 2001.

[33] ISO/IEC 15938. Information Technology – Multimedia Content Description Interface. First Edition,2002.

[34] Data modeling. http://en.wikipedia.org/wiki/Data_modeling (consulted on Aug.10, 2006)

[35] Guojun Lu. Multimedia Database Management Systems, Artech House Publishers, INC, 1999.

[36] Gupta, A., Weymouth, T., and Jain, R., 'Semantic Queries with Pictures: The VIMSYS Model,' Proceedings of the 17th Conference on Very Large Databases, Palo Alto, California (1991), pp. 69-79.

[37] Gudivada, V., Raghavan, V., and Vanapipat, K., 'A Unified Approach to Data Modeling and Retrieval for a Class of Image Database Applications,' IEEE Transactions on Data and Knowledge Engineering, (1994).

[38] R. Chbeir, S. Atnafu, L. Brunie, Image Data Model for Efficient Multi-Criteria Query in Medical Database; 14th International Conference on Scientific and Statistical Database Management (SSDBM), Edinburgh, Scotland, 24th-26th July 2002, pp.165- 175.

[39] Zhong, D. and Chang, S. –F. An integrated approach for content-based video object segmentation and retrieval. IEEE Transactions on Circuits and Systems for Video Technology, December 1999, vol. 9(8), 1259-1268.

[40] Rui, Y., Huang, S. and Mehrotra, S. Constructing table-of-content for videos. ACM Springer-Verlag Multimedia Systems Journal, 1999, vol. 7(5), 409-423.

[41] Oh, J. and Hua, K. A. An efficient and cost-effective technique for browsing, querying and indexing large video databases. In Proc. of ACM Int'l Conference on Management of Data (SIGMOD), Dallas, USA, May 2000.

[42] Swanberg, D., Shu, C. –F. and Jain, R..Knowledge guided parsing in video databases. In Proc. of IS&T/SPIE Conference on Image and Video Processing, San Jose, CA, February 1993, vol. 1908, 13-24.

[43] Smith, T. G. A and Davenport, G. The stratification system. A design environment for random access video. In Proceedings of the 3rd Int'l Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, CA, 1992.

[44] Jiang, H., Montesi, D. and Elmagarmid, K. Videotext database systems. In Proc. Of IEEE Int'l Conference on Multimedia Computing and Systems, Ontario, Canada, June 1997.

[45] Dessalegn Mekuanint, Similarity-based video retrieval: Modeling and processing. MSc. Thesis, Addis Ababa University, June 2004.

[46] Y.Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis using both audio and visual clues," IEEE Signal Processing Mag., vol. 17, no. 6, pp. 12–36, 2000.

[47] R. M. Aarts, and R. T. Dekkers. A real-time speech-music discriminator. J. Audio Eng. Soc.,1999, 47(9).

[48] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. Speech music discrimination for multimedia applications. In ICASSP, , 2000, vol. IV, pages 2445-2448.

[49] Musclefish homepage, available on: http://www.musclefish.com (consulted on July 16, 2006)

[50] Chrisitan Spevak, Emmanuel Favreau : SoundSpotter- A prototype system for content-based audio retrieval. In Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02), Hamburg, Germany, September 26-28, 2002.

[51] Allamanche E., Herre J., Hellmuth O., Bernhard Fröbach B. and Cremer M., "AudioID: Towards Content-Based Identification of Audio Material", 100th AES Convention, Amsterdam, Netherlands, May 2001.

[52] Neuschmied H., Mayer H. and Battle E., "Identification of Audio Titles on the Internet", Proceedings of International Conference on Web Delivering of Music, Florence, Italy, November 2001.

[53] Fragoulis D., Rousopoulos G., Panagopoulos T., Alexiou C. and Papaodysseus C., “On the Automated Recognition of Seriously Distorted Musical Recordings”, IEEE Transactions on Signal Processing, April 2001, vol.49, no.4, p.898-908.

[54] J. Goldman, S. Renals, S. Bird, F. de Jong M. Federico, C. Fleischhauer, M. Kornbluh, L. Lamel, Accessing the spoken word, International journal on digital libraries, V. 5(4), pp. 287-298, Aug. 2005.

[55] Elizabeth Shriberg. “To `errrr' is human: ecology and acoustics of speech disfluencies”. Journal of the International Phonetic Association, 2001, 31(1): pp.153-169.

[56] S.E. Johnson, P. Jourlin, K. Spark Jones and P.C. Woodland Spoken Document Retrieval for TREC-8 at Cambridge University, Proc. TREC-8, 2000, pp. 197-206.

[57] Pedro Cano , Markus Koppenberger , Sylvain Le Groux, Perfecto Herrera, Julien Ricard and Nicolas Wack: Knowledge and Content-based Audio Retrieval using WordNet. ICETE(3),2004,pp. 301-308.

[58] J. Garfolo, E. Voohees, C. Auzanne, V.Stanford and B. Lund 1998 TREC-7 Spoken Document Retrieval Track Overview and Results Proc. TREC-7, NIST SP 500-242, 1999, pp.79-90.

[59] J. Garfolo, C. Auzanne and E. Voohees 1999 TREC-8 Spoken Document Retrieval Track: Overview Results and Analysis To appear in Proc. TREC-8

[60] Wei-Ta Chu, Wen-Huang Cheng, Jane Yung-Jen Hsu and Ja-LingWu: Toward semantic indexing and retrieval using hierarchical audio models. Multimedia Systems ,2005, 10(6): 570–583.

[61] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. In Proc. 2002 IEEE International Conference on Multimedia and Expo, 2002, volume 1, pages 345–348.

[62] F. Pachet. Musical metadata and knowledge management. In Encyclopedia of knowledge Management, Idea Group, 2005, pp 672–677.

[63] Oracle (2003). Oracle interMedia Reference, 10g Release 1. available at: http://download-east.oracle.com/docs/ (consulted on Nov. 23, 2006).

[64] IBM (2001). DataBlade Module Development Overview, Version 4.0. available at: http://www-306.ibm.com/software/data/informix/blades/ (consulted on Nov.23,2006)

[65] Road Ward. Oracle interMedia User’s Guide and Reference, Release 9.0.1, Oracle Corporation, Part No. A88786-01, 2001.

[66] M. Hel´en and T. Lahti, “Query-by-Example Methods for Audio Signals,” in Proc. 7th IEEE Nordic Signal Processing Symposium, Reykjavik, Iceland, June 2006, pp. 302–305.

[67] J.A. Haitsma and T. Kalker, A Highly Robust Audio Fingerprinting System, Proc. ISMIR 2002, Paris, October 2002.

[68] ISO/IEC 15938. Information Technology – Multimedia Content Description Interface. First Edition ,2002.

[69] Benetos, E., Kotti, M., Kotropoulos, C., Burred, J., Eisenberg, G., Haller, M., & Sikora, T. Comparison of Subspace Analysis-Based and Statistical Model-Based Algorithms for Musical Instrument Classification. 2nd Workshop on Immersive Communication and Broadcast Systems (ICOB),2005.

[70] Xiong, Z., Radhakrishnan, R., Divakaran, A., Huang, S.T. Comparing MFCC and MPEG-7 audio features for feature extraction, Maximum Likelihood HMM and Entropic Prior HMM for sports audio classification. Proceedings of the International Conference on Multimedia and Expo,2003, vol. 3. 397-400.

[71] MPEG-7 Audio Analyzer Low-Level Descriptors Extractor, available at: http://mpeg7lld.nue.tu-berlin.de. (consulted on March, 21, 2006).

[72] Eric Allamanche, J¨urgen Herre, Oliver Hellmuth, Bernhard Fr¨oba, and Markus Cremer. AudioID: Towards Content-Based Identification of Audio Material. In 110th AES-Convention, Amsterdam, Convention Paper 5380, 2001.

[73] Oliver Hellmuth, Eric Allamanche, J¨urgen Herre, Thorsten Kastner, Markus Cremer, and Wolfgang Hirsch. Advanced Audio Identification using MPEG-7 Content Description. In 111th AESConvention, New York, Convention Paper 5463, 2001.

[74] D. Mitrovic, M. Zeppelzauer, H. Eidenberger. Analysis of the Data Quality of Audio Features of Environmental Sounds. J.UKM (Journal of Universal Knowledge Managemnet). 1(1), 2006.

[75] S. Doraisamy and S. Rüger. Robust Polyphonic Music Retrieval with N-grams. Journal of Intelligent Information Systems, Springer Netherlands, 21(1), July, 2003

[76] J.M. Martinez. MPEG-7 Overview (version10.0). ISO/IEC JTC1/SC29/WG11N6828, MPEG Requirements Group, Palma de Mallorca, October 2004. Available at: http://www.chiariglione.org/mpeg /standards/mpeg-7/mpeg-7.htm. (Consulted on Jul.20, 2006).

[77] Sensitivity of a human ear. Available on the site: http://hyperphysics.phyastr.gsu.edu /hbase/sound/earsens.html. (Consulted on Sept. 21, 2006).

DECLARATION

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials used for this thesis have been duly acknowledged.

Declared by:

Name: ____Tizeta Zewide___

Signature: _____

Date: _____

Advisor:

Name: Solomon Atnafu (Ph.D)

Signature: _____

Date: _____