Multimedia Tools and Applications, 15, 269–290, 2001

Indexing and Retrieval of Audio: A Survey

GUOJUN LU guojun.lu@infotech.monash.edu.au
Gippsland School of Computing and Information Technology, Monash University, Churchill 3842, Australia

Abstract. With more and more audio being captured and stored, there is a growing need for automatic audio indexing and retrieval techniques that can retrieve relevant audio pieces quickly on demand. This paper provides a comprehensive survey of audio indexing and retrieval techniques. We first describe main audio characteristics and features and discuss techniques for classifying audio into speech and music based on these features. Indexing and retrieval of speech and music is then described separately. Finally, significance of audio in multimedia indexing and retrieval is discussed.

Keywords: audio indexing and retrieval, audio classification, multimedia indexing and retrieval, audio features, information retrieval

1. Introduction

With the advances of information technology, more and more digital audio, images and video are being captured, produced and stored. There have been strong research and development interests in multimedia indexing and retrieval in order to effectively and efficiently use the information stored in these media types. Many publications have focused on image and video indexing and retrieval [1, 2, 19, 32]. This paper attempts to provide a comprehensive survey of audio indexing and retrieval techniques.

Human beings have amazing ability to distinguish different types of audio. Given any audio piece, we can instantly tell the type of audio (e.g., human voice, music or noise), speed (fast or slow), the mood (happy, sad, relaxing etc.), and determine its similarity to another piece of audio. However, a computer sees a piece of audio as a sequence of sample values. At the moment, the most common method of accessing audio pieces is based on their titles or file names. Due to the incompleteness and subjectiveness of the file name and text description, it may be hard to find audio pieces satisfying the particular requirements of applications. In addition, this retrieval technique cannot support queries such as "find audio pieces similar to the one being played" (query by example).

To solve the above problems, content based audio retrieval techniques are required. The simplest content-based audio retrieval uses sample to sample comparison between the query and the stored audio pieces. This approach does not work because audio signals are variable and different audio pieces may be represented by different sampling rates and may use a different number of bits for each sample. Because of this, content based audio retrieval is commonly based on a set of extracted audio features, such as frequency distribution.

The following general approach to content based audio indexing and retrieval is normally taken.

- Firstly, audio is classified into some common types of audio such as speech, music and noise.
- Secondly, the different audio types are processed and indexed in different ways. For example, if the audio type is speech, speech recognition is applied and the speech is indexed based on recognized words.
- Thirdly, query audio pieces are similarly classified, processed and indexed.
- Fourthly, audio pieces are retrieved based on similarity between the query index and the audio indices in the database.

The audio classification step is important for several reasons. Firstly, different audio types require different processing and indexing retrieval techniques. Secondly, different audio types have different significance to different applications. Thirdly, one of the most important audio types is speech and there are now quite successful speech recognition techniques/systems available. Fourthly, the audio type or class information is itself very useful to some applications. Fifthly, the search space after classification is reduced to a particular audio class during the retrieval process. There are some works that do not classify audio into different types before carrying out indexing and retrieval based on transformed audio sample values [33]. But the reported retrieval accuracy is very low. Thus we will not discuss them further in this paper.

Audio classification is based on some objective or subjective audio features. Thus before we discuss audio classification in Section 3, we describe a number of major audio features in Section 2. In our discussion, we assume audio files are in uncompressed form.

One of the major audio types is speech. The general approach to speech indexing and retrieval is to first apply speech recognition to convert speech to spoken words and then apply traditional information retrieval techniques (IR) on the recognized words [7, 26]. Thus speech recognition techniques are critical to speech indexing and retrieval. Section 4 discusses the main speech recognition techniques.

There are two forms of musical representation: structured and sample-based. We describe general approaches to the indexing and retrieval of music in Section 5.

In some applications, a combination of multiple media types are used to represent information (multimedia objects). We can use the temporal and content relationships between different media types to help with the indexing and retrieval of multimedia objects. We briefly describe this in Section 6.

Section 7 summarizes this paper.

2. Main audio properties and features

In this section, we describe a number of common features of audio signals. These features are used for audio classification and indexing in later sections. Audio perception is itself a complicated discipline. A complete coverage of audio features and their effects on perception is beyond the scope of this paper. Interested readers are referred to [3, 17].

Audio signals can be represented in the time domain (time-amplitude representation) or the frequency domain (frequency-magnitude representation). Different features can be derived or extracted from these two representations. In the following, we describe features

obtained in these two domains separately. In addition to features that can be directly calculated in these two domains, there are other subjective features such as timbre. We will briefly describe these features too.

2.1. Features derived in the time domain

Time domain or time-amplitude representation is the most basic signal representation technique, where a signal is represented as amplitude varying with time. Figure 1 shows a typical digital audio signal in the time domain. In the figure, silence is represented as 0. The signal value can be positive or negative depending on whether the sound pressure is above or below the equilibrium atmospheric pressure when there is silence. It is assumed that 16 bits are used for representing each audio sample. Thus the signal value ranges from 32767 $(2^{15} - 1)$ to -32767.

From the above representation, we can easily obtain the average energy, zero crossing rate and silence ratio.

2.1.1. Average energy. The average energy indicates the loudness of the audio signal. There are many ways to calculate it. One simple calculation is as follows:

$$E = \frac{\sum_{n=0}^{N-1} x(n)^2}{N}$$

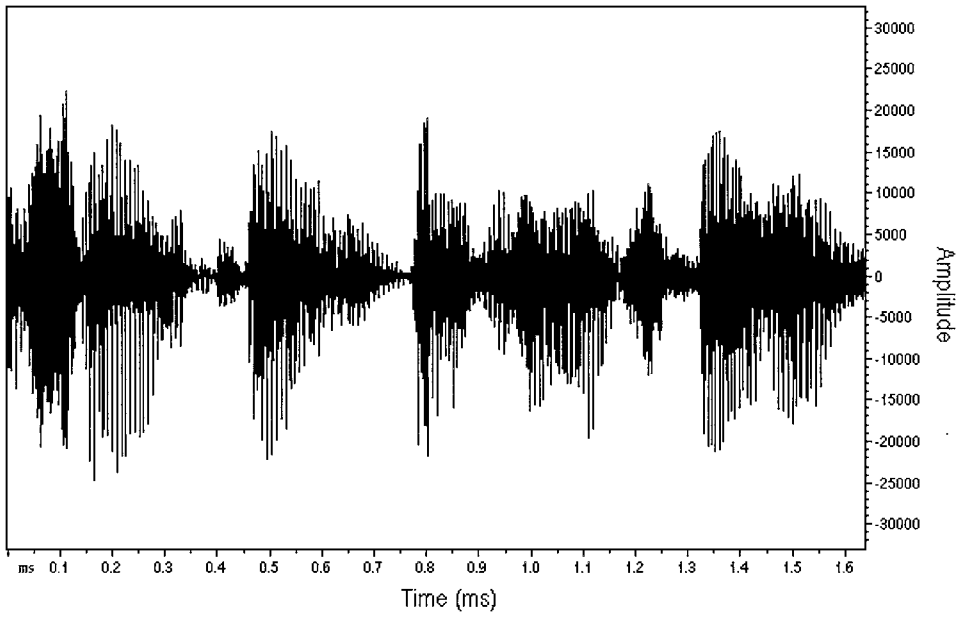


Figure 1. Amplitude-time representation of an audio signal.

where E is the average energy of the audio piece, N is the total number of samples in the audio piece, and $x(n)$ is the sample value of sample n.

2.1.2. Zero crossing rate. The zero crossing rate indicates the frequency of signal amplitude sign change. To some extent, it indicates the average signal frequency. The average zero crossing rate is calculated as follows:

$$ZC = \frac{\sum_{n=1}^{N} |\text{sgn}\, x(n) - \text{sgn}\, x(n-1)|}{2N}$$

where sgn $x(n)$ is the sign of $x(n)$ and will be 1 if $x(n)$ is positive and -1 if $x(n)$ is negative.

2.1.3. Silence ratio. The silence ratio indicates the proportion of the sound piece that is silent. Silence is defined as a period within which the absolute amplitude values of a certain number of samples are below a certain threshold. Note that there are two thresholds in the definition. The first is used to determine if an audio sample is silent. But an individual silent sample will not be considered as a silent period. Only when the number of consecutive quiet samples is above a certain time threshold are these samples considered to make up a silent period. The silence ratio is calculated as the ratio between the sum of silent periods and the total length of the audio piece.

The critical issue is how to decide if a sample is silent. There are three common approaches to this problem. In the first approach, a fixed amplitude threshold is selected and any sample whose value is below the threshold is considered as silent. The main advantage of this approach is that it is easy to implement. But the weakness of this approach is that it is difficult to select a single silent threshold for all audio pieces as the amplitude of a signal depends on recording conditions. For example, different background noises may change the value of silent samples.

The second approach is to select a reference silence value for each audio piece and any sample whose amplitude is below its reference silence value is considered as silent. The advantage of this approach is its efficiency in short audio pieces, its disadvantage is that it is not suited for signals in which there is an evolution of sound signal, for example, there is a variation of background sound.

In the third approach, audio pieces are divided into small time intervals. Adaptive silence thresholds are used which vary from interval to interval depending on the sample statistics of each time interval. This approach is effective but complicated to implement.

2.2. Features derived from the frequency domain

2.2.1. Sound spectrum. The time domain representation does not show the frequency components and frequency distribution of a sound signal. These can be represented in frequency domain. The frequency domain representation can be derived from the time domain representation according to the Fourier Transform. The Fourier Transform can be loosely stated as that any signal can be decomposed into its frequency components. In the frequency domain, the signal is represented as amplitude varying with frequency, indicating

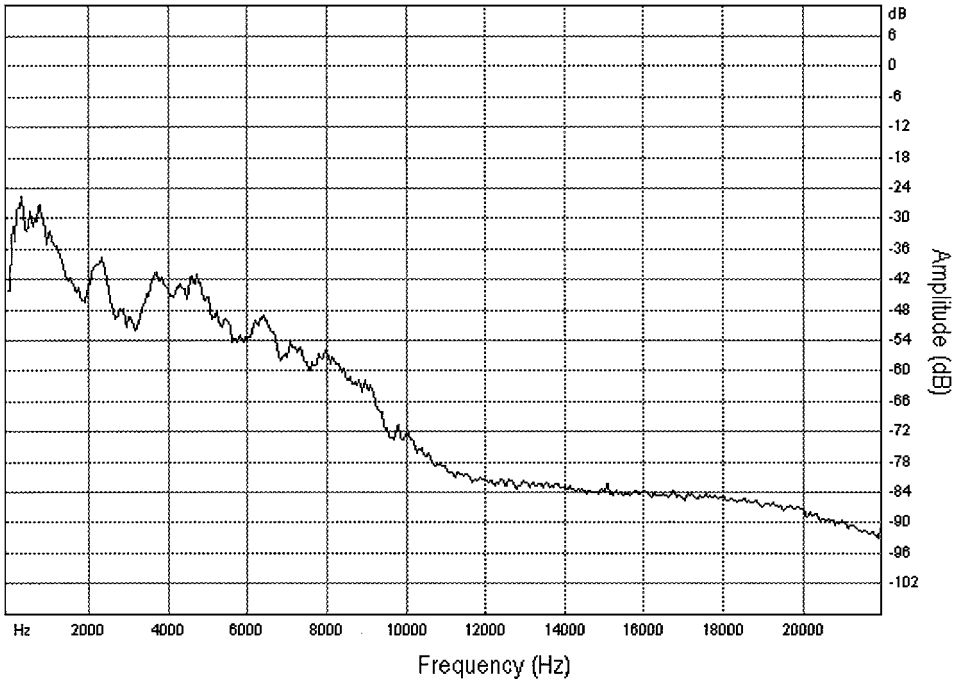


Figure 2. The spectrum of the sound signal in figure 1.

the amount of energy at different frequencies. The frequency domain representation of a signal is called the spectrum of the signal. We look at an example spectrum first and then briefly describe how the spectrum is obtained using the Fourier Transform.

Figure 2 shows the spectrum of the sound signal of figure 1. In the spectrum, frequency is shown on the abscissa and amplitude is shown on the ordinate. From the spectrum, it is easy to see the energy distribution across the frequency range. For example, the spectrum in figure 2 shows that most energy is in the frequency range 0 to 10 kHz.

Now let us see how to derive the signal spectrum, based on the Fourier Transform. As we are interested in digital signals, we use the Discrete Fourier Transform (DFT), given by the following formula:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jn\omega_k}$$

where $\omega_k = 2\pi k/N$, $x(n)$ is a discrete signal with N samples, k is the DFT bin number.

If the sampling rate of the signal is f_s Hz, then the frequency f_k of bin k in Hz is given by:

$$f_k = f_s \frac{\omega_k}{2\pi} = f_s \frac{k}{N}$$

where $k \leq \frac{N}{2}$.

If $x(n)$ is time-limited to length N, then it can be recovered completely by taking the Inverse Discrete Fourier Transform (IDFT) of the N frequency samples as follows:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jn\omega_k}$$

The DFT and IDFT can be calculated efficiently using algorithms called Fast Fourier Transforms (FFT).

As stated above, the DFT operates on finite length (length N) discrete signals. In practice, many signals extend over a long time period. It would be difficult to do a DFT on a signal with very large N. To solve this problem, the Short Time Fourier Transform (STFT) was introduced. In the STFT, a signal of arbitrary length is broken into blocks called *frames* and the DFT is applied to each of the frames. Frames are obtained by multiplying the original signal with a window function. We will not go into details of the STFT here. Interested readers are referred to [13, 21, 23]. Typically, a frame length of 10 ms to 20 ms is used in sound analysis.

In the following, we describe a number of features that can be derived from the signal spectrum.

2.2.2. Bandwidth. The bandwidth indicates the frequency range of a sound. Music normally has a higher bandwidth than speech signals. The simplest way of calculating bandwidth is by taking the frequency difference between the highest frequency and lowest frequency of the non-zero spectrum components. In some cases "non-zero" is defined as at least 3 dB above the silence level.

2.2.3. Energy distribution. From the signal spectrum, it is very easy to see the signal distribution across the frequency components. For example, we can see if the signal has significant high frequency components. This information is useful for audio classification because music normally has more high frequency components than speech. So it is important to calculate low and high frequency band energy. The actual definitions of "low" and "high" are application dependent. For example, we know that the frequencies of a speech signal seldom go over 7 kHz. Thus we can divide the entire spectrum along the 7 kHz line: frequency components below 7 kHz belong to the low band and others belong to the high band. The total energy for each band can be calculated as the sum of power of each samples within the band.

One important feature that can be derived from the energy distribution is the spectral *centroid*, which is the midpoint of the spectral energy distribution of a sound. Speech has low centroid compared to music. The centroid is also called *brightness*.

2.2.4. Harmonicity. The second frequency domain feature of the sound is harmonicity. In harmonic sound the spectral components are mostly whole number multiples of the lowest, and most often loudest frequency. The lowest frequency is called *fundamental frequency*. Music is normally more harmonic than other sounds. Whether a sound is harmonic can be determined by checking if the frequencies of dominant components are of multiples of the fundamental frequency.

For example, the sound spectrum of the flute playing the note G4 has a series of peaks at frequencies of

400 Hz, 800 Hz, 1200 Hz, 1600 Hz, etc.

We can write the above series as

$f, 2f, 3f, 4f,$ etc.

where $f = 400$ Hz is the fundamental frequency of the sound. The individual components with frequencies of nf are called *harmonics* of the note.

Note that the fundamental frequency may not exist in some signals. In this case, whether a sound is harmonic can be determined by checking if the frequencies of dominant components are of multiple of a common lower frequency (the missing fundamental).

2.2.5. Pitch. The third frequency domain feature is pitch. Only period sounds, such as those produced by musical instruments and the voice, give rise to a sensation of pitch. Sounds can be ordered according to the levels of pitch. Most percussion instruments, as well as irregular noise, don't give rise to a sensation by which they could be ordered. Pitch is a subjective feature, which is related to but not equivalent to the fundamental frequency. However, in practice, we use the fundamental frequency as the approximation of the pitch.

2.3. Spectrogram

The amplitude-time representation and spectrum are two simplest signal representations. Their expressive power is limited in that the amplitude-time representation does not show the frequency components of the signal and the spectrum does not show when the different frequency components occur. To solve this problem, a combined representation called a spectrogram is used. The spectrogram of a signal shows the relation between three variables: frequency content, time and intensity. In the spectrogram, frequency content is shown along the vertical axis, and time along the horizontal one. The intensity, or power, of different frequency components of the signal is indicated by a gray scale, the darkest part marking the greatest amplitude/power.

Figure 3 shows the spectrogram of the sound signal of figure 1. The spectrogram clearly illustrates the relationships among time, frequency and amplitude. For example, we can see from figure 3 that there are two strong high frequency components of up to 8 kHz appearing at 0.7 ms and 12.3 ms.

We can determine the regularity of occurrence of some frequency components from the spectrogram of a signal. Music spectrogram is more regular.

2.4. Subjective features

Except for pitch, all of the features described above can be directly measured in either the time domain or the frequency domain. There are other features that are normally subjective. One such feature is timbre.

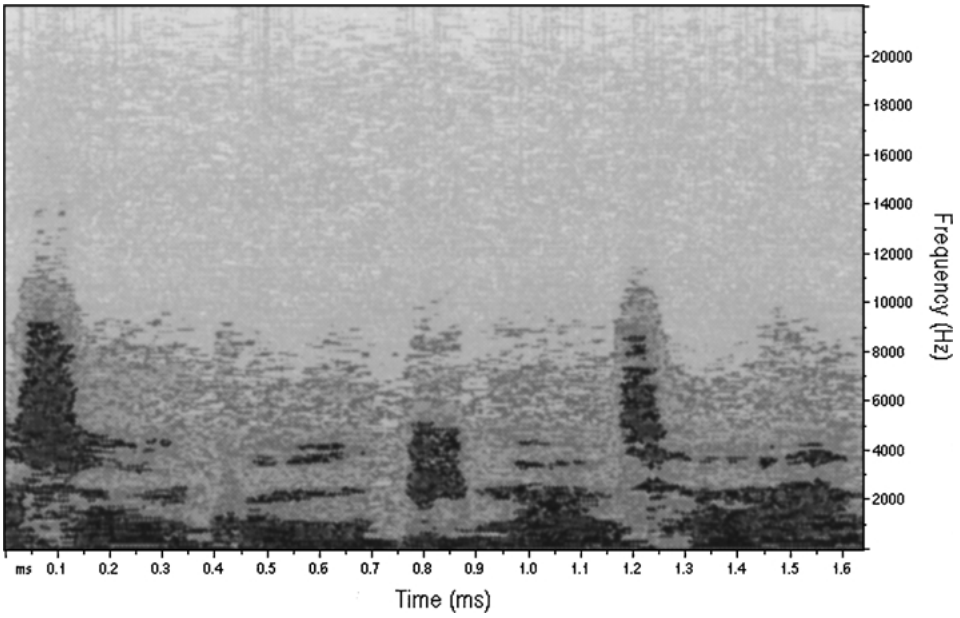


Figure 3. The spectrogram of the sound signal of figure 1.

Timbre is not well understood and defined. It encompasses all the distinctive qualities of a sound other than its pitch, loudness and duration. Salient components of timbre include the amplitude envelope, harmonicity, and spectral envelope.

3. Audio classification

We have mentioned five reasons why audio classification is important in Section 1. In this section, we first summarize the main characteristics of different types of sound, based on the features described in the previous section. We broadly consider two types of sound—speech and music, although each of these sound types can be further divided into different sub-types such as male and female speech, and different types of music. We then present two types of classification frameworks and their classification results. There are other types of sound such as noise and various sound effects. The characteristics of these types of sound vary greatly and are difficult to generalize. They can only be identified in specific domains. Thus, we will not discuss these types of sound further in this paper.

3.1. Main characteristics of different types of sound

In the following we summarize the main characteristics of speech and music. They are the basis for audio classification.

3.1.1. Speech. The bandwidth of a speech signal is generally low compared to music. It is normally within the range 100 to 7000 Hz. Because speech has mainly low frequency components, the spectral centroids (also called brightness) of speech signals are usually lower than those of music.

There are frequent pauses in a speech, occurring between words and sentences. Therefore, speech signals normally have higher a silence ratio than music.

The characteristic structure of speech is a succession of syllables composed of short periods of friction (caused by consonants) followed by longer periods for vowels [27]. It was found that during the fricativity, the average zero-crossing rate (ZCR) rises significantly. Therefore, compared to music, speech has higher variability in ZCR.

3.1.2. Music. Music normally has a high frequency range, from 0 to 20000 Hz. Thus, its spectral centroid is higher than that of speech.

Compared to speech, music has a lower silence ratio. One exception may be music produced by a solo instrument or singing without accompanying music.

Compared to speech, music has lower variability in ZCR.

Music often has regular beats which can be extracted to differentiate it from speech [28].

Table 1 summarize the major characteristics of speech and music. Note that the list is not exhaustive. There are other characteristics derived from specific characteristics of speech and music [31].

3.2. Audio classification frameworks

All classification methods are based on calculated feature values. But they differ in how these features are used. In the first group of methods, each feature is used individually in different classification steps [20, 22], while in the second group a set of features is used together as a vector to calculate the closeness of the input to the training sets [6, 31]. We discuss these two types of classification frameworks. It should be noted that although most techniques calculate features from raw audio samples (uncompressed audio data), there have been proposals to calculate some features directly from compressed audio files [20].

3.2.1. Step by step classification. In this approach to audio classification, each audio feature is used separately to determine if an audio piece is music or speech. Each feature

Table 1. Main characteristics of speech and music.

Features	Speech	Music
Bandwidth	0–7 kHz	0–20 kHz
Spectral centroid	low	high
Silence ratio	high	low
Zero-crossing rate	more variable	less variable
Regular beat	no existing	often existing

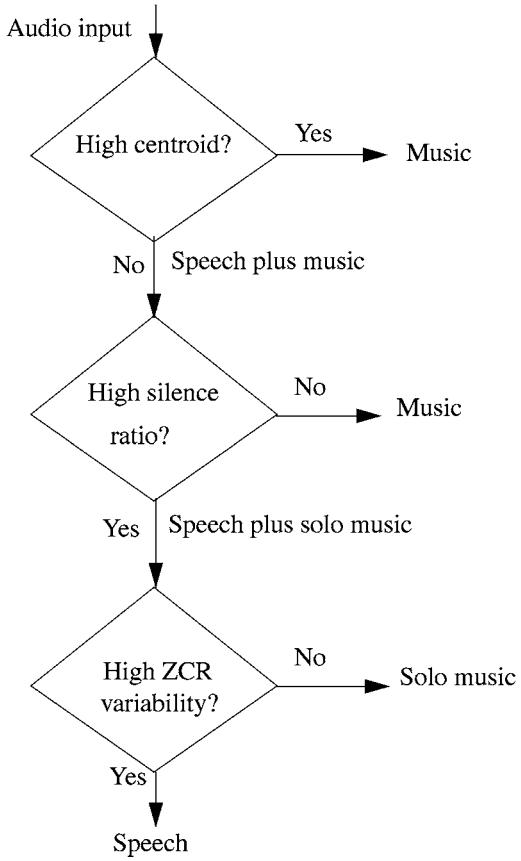


Figure 4. A possible audio classification process.

can be seen as a filtering or selection criterion. At each filtering step, an audio piece is determined as one type or another. A possible filtering process is shown in figure 4 [12]. Firstly, the centroid of all input audio pieces is calculated. If an input has a centroid higher than a preset threshold, it is deemed to be music. Otherwise, the input is speech or music because not all music has high centroid. Secondly, the silence ratio is calculated, if the input has low silence ratio, it is deemed to be music. Otherwise, the input is speech or solo music because solo music may have a very high silence ratio. Finally, we calculate ZCR. If the input has very high ZCR variability, it is speech. Otherwise, it is solo music.

In this classification approach, it is important to determine the order in which different features are used for classification. The order is normally decided based on computational complexity and the differentiating power of the different features. The less complicated feature with high differentiating power is used first. This will reduce the number of steps that a particular input will go through and reduce the total required amount of computation.

Multiple features and steps are used to improve classification performance. In some applications, audio classification can be based on only one feature. For example, Saunders

[27] used ZCR variability to discriminate broadcast speech and music and achieved an average successful classification rate of 90% [27]. Lu and Hankinson [12] used the silence ratio to classify audio into music and speech with an average success rate of 82%.

Minami et al. [16] used characteristics of sound spectrograms to detect music and speech. They noted that peaks of the spectrogram appear to be stable in the frequency direction when music is present. Therefore, music can be detected by extracting these stable peaks and calculating their durations. On the other hand, the most apparent feature of speech is the equally spaced strips caused by harmonic structure of voiced sound. So the speech can be detected by identifying these strips. It is quite common that a speech is accompanied by some background music. Minami et al. proposed to first detect music from the spectrogram and then to remove the music components from the spectrogram. They then detect if speech exists in the spectrogram with music components removed.

3.2.2. Feature vector based audio classification. In this approach to audio classification, values of a set of features are calculated and used as a feature vector. During the training stage, the average feature vector (reference vector) is found for each class of audio. During classification, the feature vector of an input is calculated and the vector distances between the input feature vector and each of the reference vectors are calculated. The input is classified into the class from which the input has least vector distance. Euclidean distance is commonly used as the feature vector distance. This approach assumes that audio pieces of the same class are located close to each other in the feature space and audio pieces of different classes are located far apart in the feature space. This approach can also be used for audio retrieval, to be discussed in Section 5.

Scheirer and Slaney [31] used 13 features including spectral centroid and ZCR for audio classification. Different features have different levels of usefulness for classification and different computation requirements. For details on this, the reader is referred to [31]. A successful classification rate of over 95% was achieved. Note that as different test sound files were used in [12, 27, 31], it is not meaningful to compare their results directly.

Compared with the step by step classification techniques, techniques based on feature vectors are theoretically more effective as multiple features are considered in the classification decision making. However, it is more computationally demanding as distances between multiple dimension feature vectors must be calculated.

In general, the duration of audio pieces has little effect on classification performance. This is because almost all techniques divide the audio piece into small windows and calculate features from these windows.

3.3. Audio segmentation

A long sound track normally consists of a mixture of speech, music and other sound types. We can use the above discussed audio classification methods to segment a long audio piece into speech and music intervals [16]. The basic idea is to divide the audio piece into a number of small windows and then apply one of the above audio classification methods to determine if the window is speech or music. Consecutive windows are then grouped into speech or music interval if they are of the same type.

4. Speech recognition and retrieval

Now that we have classified audio into speech and music, we can deal with them separately with different techniques. This section looks at speech retrieval techniques and the next section deals with music.

The basic approach to speech indexing and retrieval is to apply speech recognition techniques to convert speech signals into text and then to apply IR techniques for indexing and retrieval. In addition to actual spoken words, other information contained in speech, such as the speaker's identity and the mood of the speaker can be used to enhance speech indexing and retrieval. In the following, we describe the basic speech recognition and speaker identification techniques.

4.1. Speech recognition

In general, the automatic speech recognition (ASR) problem is a pattern matching problem. An ASR system is trained to collect models or feature vectors for all possible speech units. The smallest unit is a phoneme. Other possible units are word and phrases. During the recognition process, the feature vector of an input speech unit is extracted and compared with each of the feature vectors collected during the training process. The speech unit whose feature vector is closest to that of the input speech unit is deemed to be the unit spoken.

In this section, we first present the basic concepts of ASR and discuss a number of factors that complicate the ASR process. We then describe the speech recognition techniques based on Hidden Markov Models (HMMs), which are most popular and produce the highest speech recognition performance.

4.1.1. Basic concepts of ASR. An ASR system operates in two stages: training and pattern matching. During the training stage, features of each speech unit is extracted and stored in the system. In the recognition process, features of an input speech unit are extracted and compared with each of the stored features and the speech unit with the best matching features is taken as the recognized unit. Without losing generality, we use a phoneme as a speech unit. If each phoneme can be uniquely identified by a feature vector independent of speakers, environment and context, speech recognition would be simple. In practice, however, speech recognition is complicated by the following factors.

- Firstly, a phoneme spoken by different speakers or by the same speaker at different times produces different features in terms of duration, amplitude and frequency components. That is, a phoneme cannot be uniquely identified with 100% certainty.
- Secondly, the above differences are exacerbated by the background or environmental noise.
- Thirdly, normal speech is continuous and it is difficult to separate it into individual phonemes because different phonemes have different durations.
- Fourthly, phonemes vary with their location in a word. The frequency components of a vowel's pronunciation are heavily influenced by the surrounding consonants [4].

Because of the above factors, the earlier ASR systems were speaker dependent, required a pause between words, and could only recognize a small number of words.

The above factors also illustrate that speech recognition is a statistical process in which ordered sound sequences are matched against the likelihood that they represent a particular string of phonemes and words. Speech recognition should also make use of knowledge of the language including a dictionary of the vocabulary and a grammar of allowable word sequences.

Figure 5 shows a general model of ASR systems. The first stage is training (top part of figure 5). In this stage, speech sequences from a large number of speakers are collected. Although it is possible to carry out speech recognition from the analog speech signals, digital signals are more suitable. So these speech sequences are converted into digital format. The digitised speech sequences are divided into frames of fixed duration. The typical frame size is 10 ms. Feature vectors are then computed for each frame. Many types of features are possible, but the most popular ones are the mel-frequency cepstral coefficients (MFCCs). MFCCs are obtained by the following process.

1. The spectrum of the speech signal is warped to a scale, called the mel-scale, that represents how a human ear hears sound.
2. The logarithm of the warped spectrum is taken.
3. An inverse Fourier transform of the result of step 2 is taken to produce what is called the cepstrum.

The phonetic modeling process uses the above obtained feature vectors, a dictionary containing all the words and their possible pronunciations, and the statistics of grammar

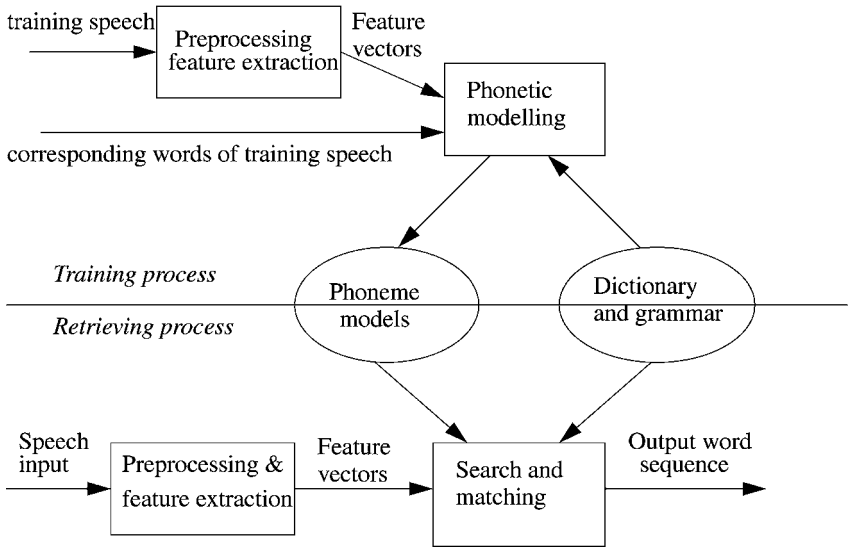


Figure 5. A general ASR system (based on [4]).

usage to produce a set of phoneme models or templates. At the end of the training stage we have a recognition database consisting of the set of phoneme models, the dictionary and grammar.

When speech is to be recognised (bottom part of figure 5), the input speech is processed in a similar way as in the training stage to produce feature vectors. The search and matching engine finds the word sequence (from the recognition database) that has the feature vector that best matches the feature vectors of the input speech. The word sequence is output as recognized text.

There are many speech recognition techniques, such as those based on dynamic time warping, HMMS, and artificial neural networks [4, 5, 18, 24, 25, 34]. Different techniques vary in features, phonetic modeling and matching methods used. In the following we describe techniques based on HMMs.

4.1.2. Techniques based on hidden Markov models. Techniques based on HMMs are currently the most widely used and produce the best recognition performance. A detailed coverage of HMMs is beyond the scope of this paper. The interested reader is referred to [24, 25] for details. In the following, we describe the basic idea of using HMMs for speech recognition.

Phonemes are fundamental units of meaningful sound in speech. They are each different from all the rest, but they are not unchanging in themselves. When one phoneme is voiced, it can be identified as similar to its previous occurrences, although not exactly the same. In addition, a phoneme's sound is modified by its neighbors' sounds which vary greatly. The challenge of speech recognition is how to model these variations mathematically.

We briefly describe what HMMs are and how they can be used to model and recognize phonemes.

A hidden Markov model consists of a number of states, linked by a number of possible transitions (figure 6). Associated with each state are a number of symbols each with a certain occurrence probability and a probability is associated with each transition. When a state is entered, a symbol is generated. Which symbol to be generated at each state is determined

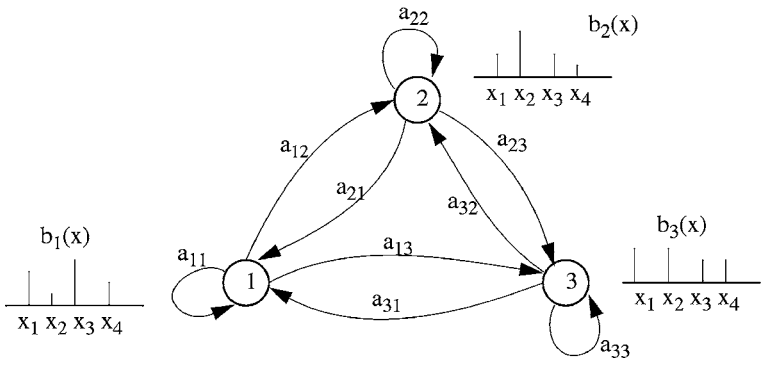


Figure 6. An example of a hidden Markov model.

by the occurrence probabilities. In figure 6, the HMM has three states. At each state, one of four possible symbols, x_1, x_2, x_3 and x_4, can be generated with different probabilities, as shown by $b_1(x)$, $b_2(x)$, $b_3(x)$ and $b_4(x)$. The transition probabilities are shown as a_{11}, a_{12}, and so forth.

In an HMM, it is not possible to identify a unique sequence of states given a sequence of output symbols. Every sequence of states that has the same length as the output symbol sequence is possible, each with a different probability. The sequence of states is "hidden" from the observer who sees only the output symbol sequence. This is why the model is called the hidden Markov model.

Although it is not possible to identify the unique sequence of state for a given sequence of output symbols, it is possible to determine which sequence of state is most likely to generate the sequence of symbols, based on state transition and symbol generating probabilities.

Now let's look applications of HMMs in speech recognition. Each phoneme is divided into three audible states: an introductory state, a middle state and an exiting state. Each state can last for more than one frame (normally each frame is 10 ms). During the training stage, training speech data are used to construct HMMs for each of the possible phonemes. Each HMM has the above three states and is defined by state transition probabilities and symbol generating probabilities. In this context, symbols are feature vectors calculated for each frame. Some transitions are not allowed as time flows forward only. For example, transitions from 2 to 1, 3 to 2 and 3 to 1 are not allowed if the HMM in figure 6 is used as a phoneme model. Transitions from a state to itself are allowed and serve to model time variability of speech.

Thus, at the end of the training stage, each phoneme is represented by one HMM capturing the variations of feature vectors in different frames. These variations are caused by different speakers, time variations and surrounding sounds.

During speech recognition, feature vectors for each input phoneme are calculated frame by frame. The recognition problem is to find which phoneme HMM is most likely to generate the sequence of feature vectors of the input phoneme. The corresponding phoneme of the HMM is deemed as the input phoneme. As a word has a number of phonemes, a sequence of phonemes are normally recognised together. There are a number of algorithms, such as forward and Viterbi algorithms, to compute the probability that an HMM generates a given sequence of feature vectors. The forward algorithm is used for recognizing isolated words and the Viterbi algorithm for recognizing continuous speech [25].

4.1.3. Speech recognition performance. Speech recognition performance is normally measured by recognition error rate. The lower the error rate, the higher the performance. The performance is affected by the following factors:

1. Subject matter: this may vary from a set of digits, a newspaper article, to general news.
2. Types of speech: read or spontaneous conversation
3. Size of the vocabulary: it ranges from tens to a few thousand words

Techniques based on HMMs perform best. It was reported at the end of 1997 that the word error rate was less than 0.3% for reading connected digits [4]. But the word error rate

was as high as 50% for general phone conversation. This shows that speech recognition performance varies greatly. For many specific applications, it is quite acceptable. However, the recognition performance for general applications is still very low and unacceptable.

4.2. Speaker identification

While speech recognition focuses on the content of speech, speaker identification or voice recognition attempts to find the identity of the speaker or to extract information about an individual from his/her speech [11]. Speaker identification is potentially very useful to multimedia information retrieval. It can determine the number of speakers in a particular setting, whether the speaker is male or female, adult or child, a person's mood, emotional state and attitude, and other information. This information, together with the speech content (derived from speech recognition) will significantly improve information retrieval performance.

Voice recognition is complementary to speech recognition. They use similar signal processing techniques to some extent. However, they differ in the following aspect. Speech recognition, if it is to be speaker-independent, must purposefully ignore any idiosyncratic speech characteristics of the speaker and focus on those parts of the speech signal richest in linguistic information. In contrast, voice recognition must amplify those idiosyncratic speech characteristics that individuate a person and suppress linguistic characteristics that have no bearing on the recognition of the individual speaker. Readers are referred to [11] for details of voice recognition.

4.3. Summary

After an audio piece is determined to be speech, we can apply speech recognition to convert the speech into text. We can then use IR techniques to carry out speech indexing and retrieval. The information obtained from voice recognition can be used to improve IR performance.

5. Music indexing and retrieval

We have discussed speech indexing and retrieval based on speech recognition in the previous section. This section deals with music indexing and retrieval. In general, research and development of effective techniques for music indexing and retrieval is still at an early stage. There are two types of music: structured or synthetic and sample based music. We briefly describe the handling of these two types of music.

5.1. Indexing and retrieval of structured music and sound effects

Structured music and sound effects are represented by a set of commands or algorithms. The most common structured music is MIDI, which represent music as a number of notes and control commands [9]. A new standard for structured audio (music and sound effects) is MPEG-4 Structured Audio, which represents sound in algorithms and control languages [29].

These structured sound standards and formats are developed for sound transmission, synthesis and production. They are not specially designed for indexing and retrieval purposes. The explicit structure and notes description existing in these formats make the retrieval process easy, as there is no need to do feature extraction from audio signals.

Structured music and sound effects are very suitable for queries requiring an exact match between the queries and database sound files. The user can specify a sequence of notes as a query and it is relatively easy to find those structured sound files that contain this sequence of notes. Although an exact match of the sequence of notes is found, it should be noted that the sound produced by the sound file may not be what the user wants because the same structured sound file can be rendered differently by different devices.

Finding similar music or sound effects to a query based on similarity instead of exact match is complicated even with structured music and sound effects. The main problem is that it is hard to define similarity between two sequences of notes. One possibility is to retrieve music based on the pitch changes of a sequence of notes [8]. In this scheme, each note (except for the first one) in the query and in the database sound files is converted into pitch change relative to its previous note. The three possible values for the pitch change are U(up), D(down) and S(same or similar). In this way, a sequence of notes is characterized as a sequence of symbols. Then the retrieval task becomes a string matching process. This scheme was proposed for sample-based sound retrieval where notes must be identified and pitch changes must be tracked with some algorithms that we will discuss in the next subsection. But this scheme is equally applicable to structured sound retrieval, where the notes are already available and pitch change can be easily obtained based on the notes scale.

5.2. Indexing and retrieval of sample-based music

There are two general approaches to indexing and retrieval of sample-based music. The first approach is based on a set of extracted sound features [36] and the second is specifically based on pitches of music notes [8, 15]. We briefly describe these two approaches separately.

5.2.1. Music retrieval based on a set of features. In this approach to music retrieval, a set of acoustic features is extracted for each sound (including queries). This set of N features is represented as an N-vector. The similarity between the query and each of the stored music pieces is calculated based on closeness between their corresponding feature vectors. This approach can be applied to general sound including music, speech and sound effects.

A good example using this approach is the work carried out at Muscle Fish LLC [36]. In this work, five features are used namely loudness, pitch, brightness, bandwidth and harmonicity. These features of sound vary over time and thus are calculated for each frame. Each feature is then represented statistically by three parameters: mean, variance and autocorrelation. The Euclidean distance or Manhattan distance between the query vector and the feature vector of each stored piece of music is used as the distance between them.

This approach can be used for audio classification, as discussed earlier. It is based on the assumption that perceptually similar sounds are closely located in the chosen feature space and perceptually different sounds are located far apart in the chosen feature space. This assumption may not be true, depending on the features chosen to represent the sound.

5.2.2. Music retrieval based on pitch. This approach is similar to pitch based retrieval of structured music. The main difference is that the pitch for each note has to be extracted or estimated in this case [8, 15]. Pitch extraction or estimation is often called pitch tracking. Pitch tracking is a simple form of automatic music transcription which converts musical sound into a symbolic representation [14, 30].

The basic idea of this approach is quite simple. Each note of music (including the query) is represented by its pitch. So a musical piece or segment is represented as a sequence or string of pitches. The retrieval decision is based on the similarity between the query and candidate strings. The two major issues are pitch tracking and string similarity measurement.

Pitch is normally defined as the fundamental frequency of a sound. To find the pitch for each note, the input music must first be segmented into individual notes. Segmentation of continuous music, especially humming and singing, is very difficult. Therefore, it is normally assumed that music is monophonic (produced using a single instrument) and stored as scores in the database. The pitch of each note is known. The common query input form is humming. To improve pitch tracking performance on the query input, a pause is normally required between consecutive notes.

There are two pitch representations. In the first method, each pitch except the first one is represented as pitch direction (or change) relative to the previous note. The pitch direction is either U(up), D(down) or S(similar). Thus, each musical piece is represented as a string of three symbols or characters.

The second pitch representation method represents each note as a value based on a chosen reference note. The value is assigned from a set of standard pitch values that is closest to the estimated pitch. If we represent each allowed value as a character, each musical piece or segment is represented as a string of characters. But in this case, the number of allowed symbols is much greater than the three that are used in the first pitch representation.

After each musical piece is represented as a string of characters, the final stage is to find a match or similarity between the strings. Considering that humming is not exact and the user may be interested in find similar musical pieces instead of just the same one, approximate matching is used instead of exact matching. The approximate matching problem is that of string matching with k mismatches. The variable k can be determined by the user of the system. The problem consists of finding all instances of a query string $Q = q_1q_2q_3 \ldots q_m$ in a reference string $R = r_1r_2r_3 \ldots r_n$ such that there are at most k mismatches (characters that are not the same). There are several algorithms that have developed to address the problem of approximate string matching [15, 36].

Both the systems of Muscle Fish LLC [36] and the University Waikato [15] produced good retrieval performance. But the performance depends on the accuracy of pitch tracking of hummed input signals. High performance is only achieved when a pause is inserted between consecutive notes.

6. Multimedia information indexing and retrieval using relationships between audio and other media

So far, we have treated sound independently of other media. In some applications, sound appears as part of a multimedia document or object. For example, a movie consists of a

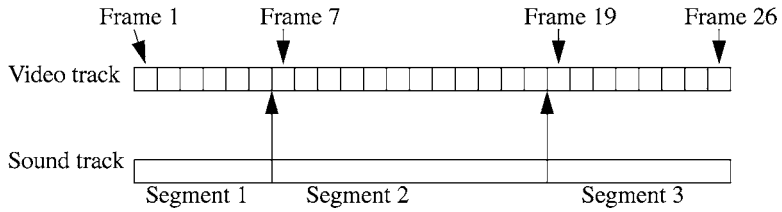


Figure 7. An example multimedia object with a video track and a sound track.

sound track and a video track with fixed temporal relationships between them. Different media in a multimedia object are interrelated in their contents as well as by time. We can use this interrelation to improve multimedia information indexing and retrieval in the following two ways [10, 20, 33, 35].

First, we can use knowledge or understanding about one medium to understand the contents of other media. We have used text to index and retrieve speech through speech recognition. We can in turn use audio classification and speech understanding to help with the indexing and retrieval of video. We use an example to show how this can be done. Figure 7 shows an multimedia object consisting of a video track and a sound track. The video track has 26 frames. Now we assume the sound track has been segmented into different sound types. The first segment is speech and corresponds to video frames 1 through 7. The second segment is loud music and corresponds to video frames 7 through 18. The final segment is speech again and corresponds to video frames 19 through 26. We can then use the knowledge of the sound track to do the following on the video track. Firstly, we can segment the video track according to the sound track segment boundaries. In this case, the video track is likely to have three segments with the boundaries alined with the sound track segment boundaries. Secondly, we can apply speech recognition to sound segments 1 and 3 to understand what was talked about. The corresponding video track may very likely have similar content. Video frames may be indexed and retrieved based on the speech content without any other processing. This is very important because in general it is difficult to extract video content even with complicated image processing techniques.

The second way to make use of relationships between media for multimedia retrieval is during the retrieval process. The user can use the most expressive and simple media to formulate a query and the system will retrieve and present relevant information to the user regardless of media types. For example, a user can issue a query using speech to describe what information is required and the system may retrieve and present relevant information in text, audio, video, or their combinations. Alternatively, the user can use an example image as query and retrieve information in images, text, audio and their combinations. This is useful because there are different levels of difficulty in formulating queries in different media.

7. Summary

This paper reviewed some common techniques and related issues for content based audio indexing and retrieval. The general approach is to classify audio into some common types such as speech and music, and then use different techniques to process and retrieve the different

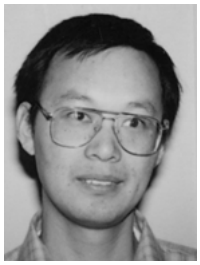
types of audio. Speech indexing and retrieval is relatively easy, by applying IR techniques on words identified using speech recognition. But speech recognition performance on general topics without any vocabulary restriction is still to be improved. For music retrieval, some useful work has been done based on audio feature vector matching and approximate pitch matching. However, more work is needed on how music and audio in general is perceived and on similarity comparison between musical pieces. It will also be very useful if we can further automatically classify music into different types such as pop and classical.

The classification and retrieval capability reviewed in this paper is potentially important and useful in many areas, such as the press and music industry, where audio information is used. For example, a user can hum or play a song and ask the system to find songs similar to the hummed or played one. A radio presenter can specify the requirements of a particular occasion and ask the system to provide a selection of audio pieces meeting these requirements. When a reporter wants to find a recorded speech, he can type in part of the speech to locate the actual recorded speech. Audio and video are often used together in situations such as movie and television programs, so audio retrieval techniques may help locate some specific video clips, and video retrieval techniques may help locate some audio segments. These relationships should be exploited to develop integrated multimedia database management systems.

References

1. P. Aigrain, H. Zhang, and D. Petkovic, "Content-based representation and retrieval of visual media: A state-of-the-art review," Journal of Multimedia Tools and Applications, Vol. 3, pp. 179–202, 1996.
2. J.R. Bach, "The virage image search engine: An open framework for image management," in Proceedings of Conference on Storage and Retrieval for Image and Video Databases IV (SPIE Proceedings Vol. 2670), 1–2 Feb., San Jose, California, 1996, pp. 76–87.
3. A.S. Bregman, Auditory Scene Analysis—The Perception Organization of Sound, The MIT Press: Cambridge, MA, 1990.
4. R. Comerford, J. Makhoul, and R. Schwartz, "The voice of the computer is heard in the land (and it listens too!)," IEEE Spectrum, Vol. 34, No. 12, pp. 39–47, 1997.
5. V. Digalakis, S. Berkowitz, E. Bocchieri, C. Boulis, W. Byrne, H. Collier, A. Corduneanu, A. Kannan, S. Khudanpur, and A. Sankar, "Rapid speech recognizer adaptation to new speakers," in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, March 15–19, Phoenix, Arizona, Vol. II, 1999, pp. 765–768.
6. J.T. Foote, "A similarity measure for automatic audio classification," in Pro. AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video and Audio Corpora, Stanford, Palo Alto, CA, Mar. 1997.
7. W.B. Frakes and R. Baeza-Yates (Eds.), Information Retrieval: Data structures and Algorithms, Prentice Hall: Englewood Cliffs, NJ, 1992.
8. A. Ghias et al., "Query by humming—Musical information retrieval in an audio database," in Proceedings of ACM Multimedia 95, November 5–9, San Francisco, California, 1995.
9. S.J. Gibbs and D.C. Tsichritzis, Multimedia Programming—Objects, Environments and Frameworks, Addison-Wesley Publishing Company: Reading, MA, 1995.
10. A.G. Hauptmann, M.J. Witbrock, A.I. Rudnicky, and S. Reed, "Speech for multimedia information retrieval," in UIST-95 Proceedings of the User Interface Software Technology Conference, Pittsburgh, Nov. 1995.
11. R.L. Klevans and R.D. Rodman, Voice Recognition, Artech House: Boston, MA, 1997.
12. G. Lu and T. Hankinson, "A technique towards automatic audio classification and retrieval," in Proceedings of International Conference on Signal Processing, Oct. 12–16, Beijing, China, 1998.

13. P.A. Lynn and W. Fuerst, Introductory Digital Signal Processing with Computer Applications, John Wiley & Sons: New York, 1989.
14. K.D. Martin, "Automatic transcription of simple polyphonic music: Robust front end processing," M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 399, 1996, available at http://sound.media.mit.edu/papers.html.
15. R.J. McNab et al., "The New Zealand digital library MELody inDex," D-Lib Magazine, May 1997, available at http://mirrored.ukoln.ac.uk/lis-journals/dlib/dlib/dlib/may97/meldex/05written.html.
16. K. Minami et al., "Enhanced video handling based on audio analysis," in Proceedings of IEEE International Conference on Multimedia Computing and Systems, June 3–6, Ottawa, Canada, 1997, pp. 219–226.
17. B.C.J. Moore, An Introduction to Psychology of Hearing, Academic Press: New York, 1997.
18. D.P. Morgan and C.L. Scofield, Neural Networks and Speech Processing, Kluwer: Dordrecht, 1991.
19. W. Niblack, X. Zhu, J.L. Hafner, T. Breuel, D.B. Panceleon, D. Petkovic, M.D. Flickner, E. Upfal, S.I. Nin, S. Sull, B.E. Dom, B.-L. Yeo, S. Srinivansan, D. Zivkovic and M. Penner, "Updates to the QBIC system," in Proceedings of Conference on Storage and Retrieval for Image and Video Databases VI (SPIE Proceedings Vol. 3312), 28–30 Jan., San Jose, California, 1998, pp. 150–161.
20. N.V. Patel and I.K. Sethi, "Audio characterization for video indexing," SPIE Proceedings, Vol. 2670, pp. 373–384, 1996.
21. A.W. Peevers, "A real time 3D signal analysis/synthesis tool based on the short time fourier transform," http://cnmat.CNMAT.Berkeley.EDU/~alan/MS-html/MSthesis.v2_ToC.html.
22. S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," http://www.informatik.uni-mannheim.de/informatic/pi4/projects/MoCA/.
23. R. Polikar, "The wavelet tutorial," http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.htm.
24. L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in Proceedings of The IEEE, Vol. 77, No. 2, 1989.
25. L.R. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall: Englewood Cliffs, NJ, 1993.
26. G. Salton and M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill: New York, 1983.
27. J. Saunders, "Real-time discrimination of broadcast speech/music," in Proceedings ACASSP'96, Vol. 2, 1996, pp. 993–996.
28. E.D. Scheirer, "Tempo and beat analysis of acoustic music signals," http://sound.media.mit.edu/~eds/papers/beat-track.html.
29. E.D. Scheirer, "The MPEG-4 structured audio standard," in Proc. IEEE ICASSP 1998, also available at http://sound.media.mit.edu/papers.html.
30. E.D. Scheirer, "Using musical knowledge to extract expressive performance information from audio recordings," available at http://sound.media.mit.edu/papers.html.
31. E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in Proceedings of the 1997 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 21–24, Munich, Germany, 1997. Also available at http://web.interval.com/papers/1996-085/index.html.
32. J.R. Smith and S.-F. Chang, "Visually searching the web for content," IEEE Multimedia Magazine, July–Sept., pp. 12–19, 1997.
33. S. Subramanya et al., "Transform-based indexing of audio data for multimedia databases," in Proceedings of IEEE International Conference on Multimedia Computing and Systems, June 3–6, Ottawa, Canada, 1997, pp. 211–218.
34. The CMU Speech Project, http://www.speech.cs.cmu.edu/speech.
35. M.J. Witbrock and A.G. Hauptmann, "Speech recognition and information retrieval," in Proceedings of the 1997 DARPA Speech Recognition Workshop, February 2–5, 1997.
36. E. Wold et al., "Content-based classification, search, and retrieval of audio," IEEE Multimedia, Vol. 3, No. 3, pp. 27–36, 1996.



Guojun Lu is currently an associate professor at Gippsland School of Computing and Information Technology, Monash University. He has held positions at Loughborough University, National University of Singapore, and Deakin University, after he obtained his Ph.D. in 1990 from Loughborough University and B.Eng. in 1984 from Nanjing Institute of Technology.

Dr. Lu's main research interests are in multimedia communications and multimedia information indexing and retrieval. He has published over 40 technical papers in these areas and wrote two books Communication and Computing for Distributed Multimedia Systems (Artech House 1996), and Multimedia Database Management Systems (Artech House 1999).