Hierarchical System for Content-based Audio Classification and Retrieval

Tong Zhang and C.-C. Jay Kuo
Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564
Email:{tzhang,cckuo}@sipi.usc.edu

ABSTRACT

A hierarchical system for audio classification and retrieval based on audio content analysis is presented in this paper. The system consists of three stages. The audio recordings are first classified and segmented into speech, music, several types of environmental sounds, and silence, based on morphological and statistical analysis of temporal curves of the energy function, the average zero-crossing rate, and the fundamental frequency of audio signals. The first stage is called the coarse-level audio classification and segmentation. Then, environmental sounds are classified into finer classes such as applause, rain, birds' sound, etc., which is called the fine-level audio classification. The second stage is based on time-frequency analysis of audio signals and the use of the hidden Markov model (HMM) for classification. In the third stage, the query-by-example audio retrieval is implemented where similar sounds can be found according to the input sample audio. The way of modeling audio features with the hidden Markov model, the procedures of audio classification and retrieval, and the experimental results are described. It is shown that, with the proposed new system, audio recordings can be automatically segmented and classified into basic types in real time with an accuracy higher than 90%. Examples of audio fine classification and audio retrieval with the proposed HMM-based method are also provided.

Keywords: audio content analysis, audio classification and retrieval, audio database, hidden Markov model, Gaussian mixture model.

1 INTRODUCTION

Audio, which includes voice, music, and various kinds of environmental sounds, is an important type of media, and also a significant part of audiovisual data. Compared to research done on content-based image and video database management, very little work has been done on the audio part of the multimedia bit stream. However, as there are more and more audio databases in place at present, people start to realize the importance of management of audio databases relying on audio content analysis.

Content-based audio classification and retrieval have a wide range of applications in the entertainment industry, audio archiving management, commercial musical usage, surveillance, etc. For example, it will be very helpful to be able to search sound effects automatically from a very large audio database in film postprocessing, which contains sounds of explosion, windstorm, earthquake, animals, and so on. There are also distributed audio libraries in the World Wide Web for management. While the use of keywords for sound browsing and retrieving provides one solution, the indexing task is however time- and labor-consuming. Moreover, an objective and consistent description of sounds may be lacking, since features of sounds are very difficult to describe. Content-based audio retrieval could be an interesting alternative for sound indexing and search. Content analysis of audio is also useful in audio-assisted video analysis. Current approaches for video indexing and retrieval mostly focus on visual

information, thus neglecting the content of the accompanying audio signal. Actually, there is an important portion of information contained in the continuous flow of audio data which often represent the theme in a simpler fashion than the visual part. For instance, all video of gun fight scenes should include the sound of shooting and/or explosion, while the image content may vary significantly from one video clip to another. This observation suggests that audio analysis can be used as the main tool for audiovisual data segmentation and indexing.

Existing research on content-based audio data management is very limited. There are in general three directions. One direction is audio segmentation and classification. One basic problem is speech/music discrimination[1], [2]. Further classification of audio may take other sounds into consideration, as done in [3], where audio was classified into "music", "speech", and "others". This work was developed for the parsing of news stories. In [4], audio recordings were classified into speech, silence, laughter, and non-speech sounds, for the purpose of segmenting discussion recordings in meetings. The second direction is audio retrieval. One specific technique in content-based audio retrieval is query-by-humming, and work in [5] gives a typical example. Two approaches for generic audio retrieval were presented, respectively, in [6] and [7]. Mel-frequency cepstral coefficients (MFCC) of audio signals were taken as features, and a tree-structured classifier was built for retrieval in [6]. It turns out that MFCC do not work well in differentiating audio timbres. In [7], statistical values (including means, variances, and autocorrelations) of several time- and frequency-domain measurements were used to represent perceptual features such as loudness, brightness, bandwidth, and pitch. This method is only suitable for sounds with a single timbre. The third direction is audio analysis for video indexing. In [8], audio analysis was applied to the distinction of five kinds of video scenes: news report, weather report, basketball game, football game, and advertisement. In [9], audio characterization was performed on MPEG sub-band level data for the purpose of video indexing.

Audio classification and retrieval is an important and challenging research topic. As described above, work in this area is still at a preliminary stage. Our objective in this research is to build a hierarchical system which consists of coarse-level and fine-level audio classificaiton and audio retrieval. There are several distinguishing features of this system. First, we divide the audio classification task into two steps. In the coarse-level step, speech, music, environmental audio, and silence are separated. This classification is generic and model-free. Then, in the fine-level step, more specific classes of natural and synthetic sounds are distinguished within each basic audio class. Second, compared with previous work, we put more emphasis on the environmental audio, which is often ignored in the past. Environmental sounds are an important ingredient in audio recordings, and their analysis is inevitable in many real applications. Third, the audio retrieval is achieved based on audio classification results, thus obtaining semantic meanings and better reliability. Irrelevant or confusing results, as often appearing in image or audio retrieval systems, are avoided by this way. Finally, we investigate physical and perceptual features of different classes of audio, and apply signal precessing techniques (including morphological and statistical analysis methods, heuristic method, clustering method, hidden Markov method, etc.) uniquely to the representation and classificaton of extracted features.

The paper is organized as follows. An overview of the proposed hierarchical system is presented in Section 2. Audio features which are important for classification and retrieval are analyzed in Section 3. Basic concepts and calculations in the Gaussian mixture model and the hidden Markov model, which are critical to the fine-level classification and retrieval methods, are introduced in Section 4. The proposed procedures for audio classification and retrieval are described in Section 5. Experimental results are shown in Section 6, and concluding remarks and future research plans are given in Section 7.

2 OVERVIEW OF PROPOSED SYSTEM

The proposed hierarchical system for audio classification and retrieval includes three stages. In the first stage, audio signals are segmented and classified into basic types, including speech, music, several types of environmental sounds, and silence. It is called the coarse-level classification. For this level, we use relatively simple features such as the energy function, the average zero-crossing rate, and the fundamental frequency to ensure the feasibility of real-time processing. We have worked on morphological and statistical analysis of these features to reveal differences among different types of audio. A rule-based heuristic procedure is built to classify audio signals based on these features. This audio coarse classification method is model-free, and can be applied under any

circumstance. It is necessary, as the first step processing of audio data, for almost any content-based audio management system. Also, an on-line segmentation and indexing of audio/video recordings is achieved based on the coarse-level classification. For example, in arranging the raw recordings of meetings or performances, segments of silence or irrelevant environmental sounds (including noise) may be discarded, while speech, music and other environmental sounds can be classified into the corresponding archives.

In the second stage, further classification is conducted within each basic type. For speech, we can differentiate it into voices of man, woman, child as well as speech with a music background. For music, we classify it according to the instruments or types (for example, classics, blues, jazz, rock and roll, music with singing and the plain song). For environmental sounds, we classify them into finer classes such as applause, bell ring, footstep, windstorm, laughter, birds' cry, and so on. This is known as the fine-level classification. Based on this result, a finer segmentation and indexing result of audio material can be achieved. Due to differences in the origination of the three basic types of audio, i.e. speech, music and environmental sounds, different approaches can be taken in their fine classification. In this work, we focus primarily on the fine classification of environmental audio. Features are extracted from the time-frequency representation of audio signals to reveal subtle differences of timbre and change pattern among different classes of sounds. The hidden Markov model (HMM) with continuous observation densities and explicit state duration densities is used as the classifier. Each kind of timbre in one audio class is represented as one state in HMM and modeled with the Gaussion mixture density. The change pattern of timbres in the audio class is modeled by the transition and duration parameters of HMM. One HMM is built for each class of sound. The fine classification of audio finds applications in automatic indexing and browsing of audio/video databases and libraries.

In the third stage, an audio retrieval mechanism is built based on the archiving scheme described above. There are two retrieval approaches. One is query-by-example, where the input is an example sound, and the output is a rank list of sounds in the database, which shows the similarity of retrieved sounds to the input query. Similar to that in the content based image retrieval system where image search can be done according to color, texture, or shape features, audio clips can also be retrieved with distinct features such as timbre, pitch, and rhythm. The user may choose one feature or a combination of features with respect to the sample audio clip. The other one is query-by-keywords (or features), where various aspects of audio features are defined in a list of keywords. The keywords include both conceptual definitions (such as violin, applause, or cough) and perceptual descriptions (such as fastness, brightness, and pitch) of sounds. In an interactive retrieval process, users may choose from a given menu a set of features, listen to retrieved samples, and modify the input feature set accordingly to get a better matched result. As the databases are organized according to audio classification schemes, audio retrieval is more efficient (for example, the retrieval may be conducted only within certain classes), and irrelevant results are avoided. Applications of audio retrieval may include searching sound effects in producing films, audio editing in making TV or radio programs, selecting and browsing materials in audio libraries, and so on.

The framework of the proposed system is shown in Figure 1. Details about features, procedures, and experimental results of the coarse-level classification and segmentation were described in our previous work [10]. In this paper, we emphasize on audio features, data models, procedures and examples for the fine-level classification and retrieval.

3 AUDIO FEATURES FOR CLASSIFICAION AND RETRIEVAL

There are two types of audio features: physical features and perceptual features. Physical features refer to mathematical measurements computed directly from the sound wave, such as the energy function, the spectrum, and the fundamental frequency. Perceptual features are subjective terms which are related to the perception of sounds by human beings, including loudness, pitch, timbre, and rhythm. For the purpose of coarse-level classification, we have used temporal curves of three kinds of short-time physical features, i.e., the energy function, the average zero-crossing rate, and the fundamental frequency. Brief concepts of these features are given below, while detailed descriptions can be found in [10]. For the fine-level classification, one of our most important tasks is to build physical and mathematical models for the perceptual features with which human beings distinguish different classes of sounds. In this work, we consider two kinds of features: timbre and rhythm.

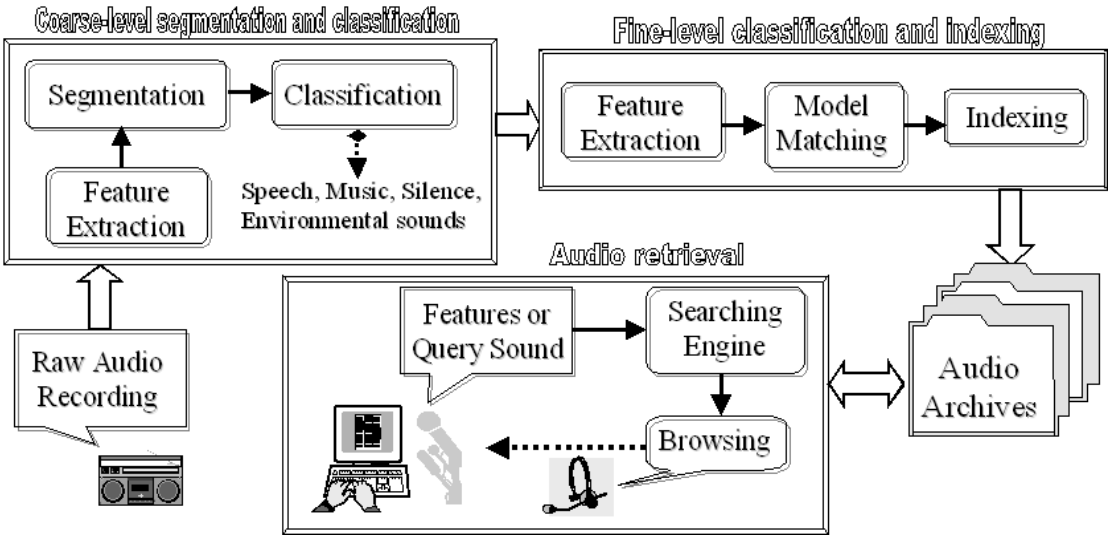


Figure 1: A hierarchical system for content-based audio classification and retrieval.

3.1 Physical features

1. *Short-time energy function.* The short-time energy of audio signal provides a convenient representation of the amplitude variation over time. For speech signals, it is a basis for distinguishing voiced speech components from unvoiced speech components, as the energy function values for unvoiced components are significantly smaller than those of the voiced components. The energy function can also be used as the measurement to distinguish silence when the SNR is high.

2. *Short-time average zero-crossing rate (ZCR).* In discrete-time signals, a zero-crossing is said to occur if successive samples have different signs. The short-time average zero-crossing rate gives rough estimates of spectral properties of audio signals. It is another measurement to differentiate voiced speech components from unvoiced speech components, as the voiced components have much smaller ZCR values than the unvoiced components. Compared to that of speech, the ZCR curve of music has a remarkablely lower variance and average amplitude. The environmenal audio of various origins can be briefly classified according to the differences in ZCR curve properties.

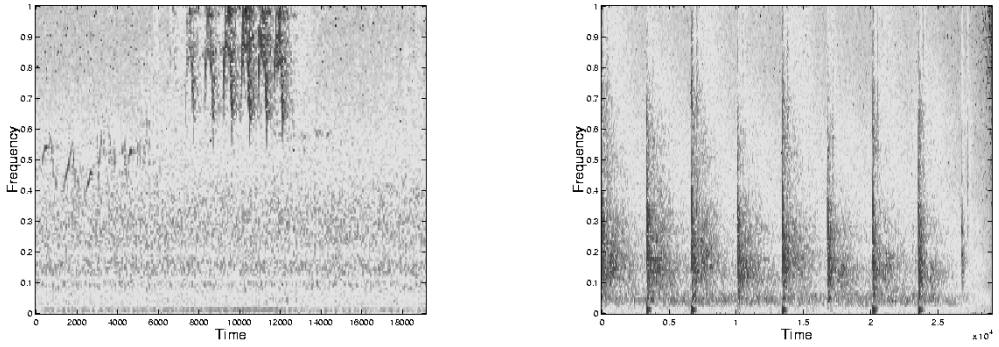
3. *Short-time fundamental frequency (FuF).* The short-time fundamental frequency reveals harmonic properties of audio signals. In the FuF curve, the amplitude is equal to the fundamental frequency when the sound is harmonic, and is set to zero when the sound is non-harmonic. Sounds from most musical instruments are harmonic. In speech, voiced components are harmonic while unvoiced components are non-harmonic. Most environmental sounds are non-harmonic except that there are some examples which are harmonic and stable, or harmonic and non-harmonic mixed.

3.2 Perceptual features

1. *Timbre.* Timbre is generally defined as "the quality which allows one to tell the difference between sounds of the same level and loudness when made by different musical instruments or voices". From the physical point of view, timbre depends primarily upon the spectrum of the stimulus. It also depends upon the waveform, the sound pressure, the frequency location of the spectrum and the temporal characteristics of the stimulus [11]. In music, it is normally believed that timbre is determined by the number and relative strengths of the instrument's partials. However, this is only close to be true [12]. The problem of building physical models for timbre perception has been investigated for a long time in psychology and music analysis without definite

answers. Nevertheless, we may get the conclusion from existing results that the temporal evolution of spectrum of audio signals acounts largely for timbre perception. We observed a large amount of various environmental sounds, and found that the timbre patterns were well reflected in the spectrograms of audio waveforms. Here, we extend timbre from a term originally used for harmonic sound (music and voice) to the perception of environmental sound, and analyze it on the time-frequency representation (such as spectrogram) of audio signals. We consider timbre as the most important feature in differentiating different classes of environmental sounds, and to build a model properly for timbre perception based on the spectrogram is one major problem in our research. Figure 2 illustrates the spectrogram of two environmental sounds. The sound shown in Figure 2(a) includes two kinds of timbres: the bird's cry (of higher frequency) and the river flow sound in the background (in lower frequency bands), which can be clearly observed from the spectrogram.

2. *Rhythm.* Rhythm is a term originally defined for speech and music. It is the quality of happening at regular periods of time. Here, we extend it to environmental sounds to represent the change pattern of timbres in a sound clip. One example is shown in Figure 2(b), where the rhythm of footstep is a signicant feature of the sound. Other sounds in which rhythm plays an important role in the perception include clock tick, telegraph machine, pager, door knock, etc.



(a) (b)

Figure 2: The spectrogram of audio signals: (a)bird-river, (b)foot-step

4 HIDDEN MARKOV MODEL AND GAUSSIAN MIXTURE MODEL

The hidden Markov model (HMM) and Gaussian mixture model (GMM) are powerful statistical tools widely used in pattern recognition. They are used to characterize the timbres and their change pattern(s) in one sound clip or a class of sounds in this work. GMM can be viewed as one component of HMM under certain circumstances.

4.1 The Gaussian Mixture Model

A Gaussian mixture density is a weighted sum of M component densities, as given by the following [13]

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}), \tag{1}$$

where $\vec{x}$ is a D-dimensional random vector, $b_i(\vec{x})$, $i = 1, \ldots, M$, are the component densities, and p_i, $i = 1, \ldots, M$, are the mixture weights. Each component density is a D-variate Gaussian function of the form

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i)\} \tag{2}$$

with mean $\vec{\mu}_i$ and covariance matrix Σ_i. The mixture weights have to satisfy the constraint $\sum_{i=1}^{M} p_i = 1$.

The complete Gaussian mixture density is parameterized by the mean vector, the covariance matrix and the mixture weight from all component densities. These parameters are collectively represented by

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\}, \quad i = 1, \ldots, M.$$

In the training process, the maximum likelihood (ML) estimation is adopted to determine model parameters which maximize the likelihood of GMM given the training data. For a sequence of T training vectors $X = \{\vec{x}_1, \ldots, \vec{x}_T\}$, the GMM likelihood can be written as

$$p(X|\lambda) = \prod_{t=1}^{T} p(\vec{x}_t|\lambda).$$

The ML parameter estimates are obtained iteratively using the expectation-maximization (EM) algorithm. At each iteration, the parameter update formulas are as below, which guarantee a monotonic increase in the likelihood value.

Mixture weight update:

$$\overline{p}_i = \frac{1}{T} \sum_{t=1}^{T} p(i|\vec{x}_t, \lambda). \tag{3}$$

Mean vector update:

$$\overline{\vec{\mu}}_i = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)\vec{x}_t}{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)}. \tag{4}$$

Covariance matrix update:

$$\overline{\Sigma}_i = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)(\vec{x}_t - \overline{\vec{\mu}}_i)(\vec{x}_t - \overline{\vec{\mu}}_i)'}{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)}. \tag{5}$$

The *a posteriori* probability for the ith mixture is given by

$$p(i|\vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\Sigma_{k=1}^{M} p_k b_k(\vec{x}_t)}. \tag{6}$$

4.2 The Hidden Markov Model

A hidden Markov model for discrete symbol observations is characterized by the following parameters [14].

1. N, the number of states in the model. We label the individual states as $\{1, 2, \ldots, N\}$, and denote the state at time t as q_t.
2. M, the number of distinct observation symbols in all states, i.e., the discrete alphabet size. We denote the individual symbols as $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$.
3. The state-transition probability distribution $A = \{a_{ij}\}$ where

   $$a_{ij} = P[q_{t+1} = j | q_t = i], \quad 1 \le i, j \le N.$$

4. The observation symbol probability distribution $B = \{b_j(k)\}$, in which

   $$b_j(k) = P[\mathbf{x}_t = \mathbf{v}_k | q_t = j], \quad 1 \le k \le M,$$

   defines the symbol distribution in state j, $j = 1, 2, \ldots, N$.
5. The initial state distribution $\pi = \{\pi_i\}$ in which

   $$\pi_i = P[q_1 = i], \quad 1 \le i \le N.$$

Thus, a complete specification of HMM includes two model parameters, N and M, the observation symbols, and the three sets of probability measures A, B, and π. We use the compact notation

$$\lambda = (A, B, \pi)$$

to indicate the complete parameter set of the model. It is used to define a probability measure for observation sequence $\mathbf{X}$, i.e., $P(\mathbf{X}|\lambda)$, which can be calculated according to a forward procedure as defined below.

Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_t, q_t = i|\lambda), \tag{7}$$

which is the probability of the partial observation sequence $\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_t$, and state i at time t, given the model λ. We can solve for $\alpha_t(i)$ inductively as follows.

1. Initialization
$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1), \ 1 \leq i \leq N. \tag{8}$$
2. Induction
$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)\alpha_{ij}] b_j(\mathbf{x}_{t+1}), \ 1 \leq t \leq T-1; \ 1 \leq j \leq N. \tag{9}$$
3. Termination
$$P(\mathbf{X}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{10}$$

4.3 HMM with Continuous Observation Densities

When observations are continuous signals/vectors, HMM with continuous observation densities should be used. In such a case, some restrictions must be placed on the form of the model probability density function (pdf) to ensure that pdf parameters can be updated in a consistent way. The most general pdf form is a finite mixture shown as follows:

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{x}, \mu_{jk}, \Sigma_{jk}), \ 1 \leq j \leq N, \tag{11}$$

where $\mathbf{x}$ is the observation vector, c_{jk} is the mixture weight for the kth mixture in state j and $\mathcal{N}$ is any log-concave or elliptically symmetric density. Without loss of generality, we assume that $\mathcal{N}$ is Gaussian with mean vector μ_{jk} and covariance matrix Σ_{jk} for the kth mixture component in state j. The mixture gains c_{jk} satisfy the stochastic constraint $\sum_{k=1}^{M} c_{jk} = 1, \ c_{jk} \geq 0, \ 1 \leq j \leq N, \ 1 \leq k \leq M$.

By comparing (11) with the Gaussian mixture density given in (1), it is obvious that the Gaussian mixture model is actually one special case of the hidden Markov model with continuous observation densities, when there is only one state in the HMM ($N = 1$) and $\mathcal{N}$ is Gaussian. The parameter update formulas in the mixture density, i.e., c_{jk}, μ_{jk}, and Σ_{jk}, are the same as those for GMM, i.e. formulas 3-6.

4.4 HMM with Explicit State Duration Density

For many physical signals, it is preferable to explicitly model the state duration density in some analytic form. That is, a transition is made only after an appropriate number of observations occur in one state (as specified by the duration density). Such a model is sometimes called the semi-Markov model. We denote the possibility of d consecutive observations in state i as $p_i(d)$. Changes must be made to the formulas for calculating $P(\mathbf{X}|\lambda)$ and updating of model parameters. We assume that the first state begins at $t = 1$ and the last state ends at $t = T$. With the forward variable $\alpha_t(i)$ now defined as

$$\alpha_t(i) = P(\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_t, \text{stay in state } i \text{ ends at } t|\lambda). \tag{12}$$

The induction steps for calculating $P(\mathbf{X}|\lambda)$ are given below.

1. Initialization

$$\alpha_1(i) = \pi_i p_i(1) \cdot b_i(\mathbf{x}_1),\ 1 \le i \le N. \tag{13}$$

2. Induction

$$\alpha_t(i) = \pi_i p_i(t) \prod_{s=1}^{t} b_i(\mathbf{x}_s) + \sum_{d=1}^{t-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_{t-d}(j) a_{ji} p_i(d) \cdot \prod_{s=t+1-d}^{t} b_i(\mathbf{x}_s),\ 2 \le t \le D,\ 1 \le i \le N. \tag{14}$$

and

$$\alpha_t(i) = \sum_{j=1}^{N} \sum_{d=1}^{D} \alpha_{t-d}(j) a_{ji} p_i(d) \prod_{s=t+1-d}^{t} b_i(\mathbf{x}_s),\ D < t \le T,\ 1 \le i \le N. \tag{15}$$

where D is the maximum duration within any state.

3. Termination

$$P(\mathbf{X}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{16}$$

5 PROCEDURES OF AUDIO CLASSIFICATION AND RETRIEVAL

5.1 Coarse-level Audio Segmentation and Classification

For on-line segmentation and classification of audio recordings, the short-time energy function, average zero-crossing rate, and fundamental frequecy are computed on the fly with incoming audio data. Whenever there is an abrupt change detected in any of these three features, a segment boundary is set. Each segment is classified into one of the basic audio types according to a rule-based heuristic procedure. The procedure includes the following steps: (1) separating silence; (2) separating environmental sounds with special features, i.e., sounds which are "harmonic and unchanged" or "harmonic and stable"; (3) distinguishing music; (4) distinguishing speech; and (5) classifying other environmental sounds to one of the following types: "periodic or quasi-periodic", "harmonic and non-harmonic mixed", "non-harmonic and stable", or "non-harmonic and irregular". Finally, a post-processing procedure is applied to reduce possible segmentation errors. For details of these processes, we refer to [10].

5.2 Fine-level Audio Classification

The core of fine-level classification is to build HMM for each class of sounds. Currently, two types of information are contained in HMM, i.e. timbre and rhythm. Each kind of timbre is modeled as one state of HMM, and represented with the Gaussion mixture density. The rhythm information is denoted by transition and duration parameters in HMM. Once HMM parameters are set, sound clips can be classified into available classes by matching to models of these classes.

5.2.1 Feature Extraction

As mentioned earlier, the timbre of sound is determined primarily by the frequency energy distribution of the sound. A key point in modeling timbre perception with HMM is the way to extract the feature vector from the short-time spectrum. Up to now, we have used the most direct way to extract features from the frequency distribution, i.e. to use the spectrum coefficients themselves. Trying to maintain a low dimension of the feature vector while at the same time keeping necessary information, we take 128-point FFT of audio signal, thus obtaining a feature vector of 65 dimensions (i.e., the logarithm of amplitude spectrum at each frequency sample between 0 and π). FFT is calculated for every 100 input samples. Therefore, for audio signals sampled at 11025Hz, there are about 110 feature vectors obtained per second for each sound.

5.2.2 Clustering

The feature vectors of one class of sounds are clustered into several sets, with each set denoting one kind of timbre, and modeled later by one state in HMM. We adopted an adaptive sample set construction method [15] for clustering with some modifications. The resulting algorithm is stated as follows.

1. Define two thresholds: t_1 and t_2, with $t_1 > t_2$.
2. Take the sample with the largest norm (denote it as $\mathbf{x}_1$) as the representative of the first cluster: $\mathbf{z}_1 = \mathbf{x}_1$, where $\mathbf{z}_1$ is the center of the first cluster.
3. Take the next sample and compute its distance to all the existing clusters $d_i(\mathbf{x}, \mathbf{z}_i)$, and choose the minimum of d_i: $\min\{d_i\}$.
   (a) If $\min\{d_i\} \leq t_2$, assign $\mathbf{x}$ to the ith cluster, and update the center of this cluster: $\mathbf{z}_i$.
   (b) If $\min\{d_i\} > t_1$, form a new cluster with $\mathbf{x}$ as the center.
   (c) If $t_2 < \min\{d_i\} \leq t_1$, do not assign $\mathbf{x}$ to any cluster, as it is in the intermediate region of clusters.
4. Repeat *Step 3* until all samples have been checked once. Calculate the variances of all the clusters.
5. If the variance is the same as last time, meaning the training process has converged, go to *Step 6*. Otherwise, return to *Step 3* for further iteration.
6. If there are still unassigned samples (in the intermediate regions), assign them to the nearest clusters. If the number of unassigned samples is larger than a certain percentage, adjust thresholds t_1 and t_2, and start with *Step 2* again.

The above procedure works well for clustering feature vectors. For example, setting $t_1 = 20$ and $t_2 = 15$, the sound of dog bark is clustered into three states: bark, intermission, and the transition period in between. Similar results were obtained with sounds of cough, footstep, etc. The sound of chime is clustered into four states corresponding to the evolution of sounds over time. Simple timbred sounds such as river flow and clock ring are clustered as having just one state. The number of states can be adjusted by changing the threshold values. As GMM is able to handle the slight differences within each state, we tend to keep the number of states as such that states have distinct differences and physical meanings.

5.2.3 Building Model

There are three cases in building HMM models for sound clips. For the first case, neither durations nor transitions of states are restricted for similar sound identification. Examples for this case include the single-state sounds and sounds such as the river with bird sound where the bird sound may happen anytime and for any length of duration upon the background river sound. For the second case, there are specific transitions among states, but the durations of states can be arbitrary. For the third case, both the duration and the transition information are critical in sound classification and retrieval, such as sounds of footstep and clock tick. The three cases have the same training process, through which a complete set of HMM parameters are obtained for each class of sounds. While during classification and retrieval, the user may choose which case is suitable for sound characterization. For the first case, only Gaussion mixture density parameters will be matched. For the second case, both GMM and transition parameters should be matched. For the third case, the whole set of HMM parameters are matched.

We denote the complete parameter set of HMM as $\lambda = (A, B, D, \pi)$, with A for the transition probability, B for GMM parameters (including mixture weights, vector means, and covariance matrices of all states), D for duration pdf parameters, and π for initial state distribution. The standard way for parameter estimation in HMM is an iterative procedure based on expectation-maximization method. However, when an explicit state duration density is included, the procedure becomes complicated with the computational load greatly increased. Besides, in such cases, there are normally fewer state transitions and much less data to estimate the duration pdf than those in standard HMM. Thus, we can simplify the procedure by breaking it into three steps.

At the first step, the observation density parameters $B = \{B_j,\ 1 \le j \le N\}$ are estimated for each state, respectively. The feature vectors in one cluster are used to train GMM parameters for that kind of timbre according to the update formulas of (3)-(6). Several implementational issues should be mentioned. First, the number of mixture components M in GMM is normally determined by experiments. In our case, we choose $M = 5$. Second, diagonal covariance matrices are selected for the ease of computation. Full covariance matrices are not necessary in GMM because the effect of using a set of full covariance Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians. Third, the initial mixture weights are random values between 0 and 1 which satisfy: $\sum_{i=1}^{M} p_i = 1$ for each state. Elements in the initial mean vectors are random values between 5 and 15, which is the concentrated range for feature vector element values. The diagonal elements in the covariance matrices are set to 1. Fourth, when there are not enough data to sufficiently train a component's variance vector or when using noise-corrupted data, the variance elements can become very small which may produce singularities in the likelihood. To avoid such singularities, a variance limiting constraint is applied as given below.

$$\overline{\sigma}_i^2 = \begin{cases} \sigma_i^2 & \text{if } \sigma_i^2 > \sigma_{\min}^2 \\ \sigma_{\min}^2 & \text{if } \sigma_i^2 \le \sigma_{\min}^2 \end{cases} \tag{17}$$

We choose $\sigma_{\min}^2 = 0.0001$. Finally, it is possible that the exponential item in (2) becomes very large (especially when the dimension of the feature vector is relatively high), and the Gaussion mixture density becomes so small that it exceeds the precision range of the computer. To keep numerical stability of the training process, a scaling factor $\exp\{c\}$ is calculated for each computation of the Gaussian mixture density, which is multiplied to every $b_i(\vec{x})$ in (1) to keep $p(\vec{x}|\lambda)$ from being too small. As shown in (6), the scaling factor is canceled out in the *a posteriori* probability so that it does not affect the parameter update. For the GMM likelihood, we can take the logarithm so that the term due to the scaling factor becomes a subtraction.

At the second step, the transition probability matrix $A = \{a_{ij}\}$ is calculated as $a_{ij} = t_{ij}/t_i,\ 1 \le i, j \le N$, where t_i is the number of transitions from state i to all other states, and t_{ij} is the number of transitions from state i to state j. The self-transition probabilities are set to 0 when explicit state duration is included, i.e. $a_{ii} = 0,\ 1 \le i \le N$.

At the third step, the duration pdf D is estimated state by state. We choose the pdf form to be the Gaussion density, i.e. $p_i(d) = \mathcal{N}(d, \mu_i, \sigma_i^2),\ 1 \le i \le N$, where μ_i and σ_i^2 are estimated statistically from the state indices of feature vectors, which are obtained through the clustering procedure. Since normally there is no restriction on which state the sound should begin with, the initial state distribution is set as $\pi_i = 1/N,\ 1 \le i \le N$. It should be noted that this simplified training procedure is not a strict HMM process. In HMM, it is unknown which vector belongs to which state (it is hidden). Here, vectors are assigned to states according to the clustering results.

5.2.4 Classification

Assume that there are K classes of sounds modeled with parameter sets $\lambda_i,\ 1 \le i \le K$. For a piece of sound to be classified, feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ are extracted. Then, the HMM likelihoods $P_i(\mathbf{X}|\lambda_i),\ 1 \le i \le K$, are computed. Choose the class j which maximizes P_i, i.e. $j = \arg\max\{P_i, 1 \le i \le K\}$, and the sound is classified into this class.

As mentioned earlier, there are three kinds of situations in matching the sound to the HMM model. For the case that the complete set of parameters are to be matched, the forward procedure described in formulas (12)-(16) are used. For the cases that the durations of states are not concerned, (7)-(10) are used to compute the likelihood with the self-transition probabilities set to 1, i.e. $a_{ii} = 1,\ 1 \le i \le N$. Furthermore, when the transition information is also not concerned, all transition probabilities are set to 1, i.e. $a_{ij} = 1,\ 1 \le i, j \le N$.

There are two problems in implementation. The first one is about the way to choose the model matching mode. During the training process, a mode index (1, 2, or 3) is assigned to each class according to the characteristics of sounds in that class. Then, the model matching mode is chosen in consistency with this index in classification. Since the way of computing the likelihood is different for different classes, there is a normalization procedure so that a comparison can be made among these likelihoods. Currently, this normalization is accomplished experimentally. An analytic solution is under our investigation. The second problem is related to numerical stability. It can be seen from the forward procedure that as t becomes large, each term of $\alpha_t(i)$ starts to approach to zero exponentially.

Two elements are inserted into the computation of $P(\mathbf{X}|\lambda)$ to keep variables from exceeding the precision range of the computer. One is to multiply each term with a scaling factor and the other is to take the logarithm of each term. Since there are addition operations in the formulas, the process is a little bit more complicated than in the training procedure.

5.3 Audio Retrieval

HMM is built for each sound clip in the audio database in the query-by-example audio retrieval. With an input query sound, its feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ are extracted, and the possibilities $P(\mathbf{X}|\lambda_i),\ 1 \leq i \leq L$ are computed according to the forward procedures, where λ_i denotes the HMM parameter set for the ith sound clip and L is the number of sound clips in the database. The user will choose, according to the characteristics of the query sound, the model matching mode and apply it to the matching of the input query to every sound in the database. A rank list of audio samples in terms of similarity with the input query will be obtained by comparing values of $P(\mathbf{X}|\lambda_i)$.

6 EXPERIMENTAL RESULTS

6.1 Audio Database and Coarse-level Classification Result

We have built a generic audio database which includes around 1500 pieces of sound of various types to test the classification and retrieval algorithms. We also collect dozens of longer audio clips recorded from movies to test the segmentation performances. The proposed coarse-level classification scheme achieves an accuracy rate of more than 90% with this audio database. Misclassification usually occurs in the hybrid sound which contains more than one basic type of audio. When testing with movie audio recordings, the segmentation and classification together can be achieved in real time. The boundaries are set accurately and each segment is properly classified. One such example can be found in [10].

6.2 Example of Fine Classification

For a brief test of the fine classification algorithm, we built the HMM parameter set for ten classes of sounds, including applause, birds' cry, dog bark, explosion, foot step, laugh, rain, river flow, thunder, and windstorm. Feature vectors extracted from 6-8 sound clips were used for building the model for each class. Then, fifty sound clips (with five pieces of sound in each class) were used to test the classification accuracy. Within the test set, most were new sound clips, while there were also some clips taken from the training set due to the lack of the sample sound in certain classes. It turned out that 41 out of the 50 sound clips were correctly classified, achieving an accuracy rate of over 80%. Misclassification happened with classes with percetually similar sounds, such as applause, rain, river, and windstorm.

6.3 Example of Audio Retrieval

In an experiment of audio retrieval, 100 short pieces of sound from 15 classes were selected to form a small database, with the HMM parameter set trained for each piece of sound. Then, we chose a sound clip of applause as the query sound, and matched it to each of the 100 HMMs. The resulting top ten sounds in the rank list belonged to the following classes: no.1-5: applause; no.6: rain; no.7-9: applause; no.10: rain. This result is reasonable, because the pouring rain and applause by a crowd of people sometimes sound alike. For another example, a sound clip of plane taking off was used as the input query, and the top ten retrieved sounds were: no.1-6: plane; no.7-10: rain. There were only 6 pieces of plane sound in the database, and they were ranked at the first 6 places, while the rest 4 places were taken by sounds of large rain.

7 CONCLUSION AND EXTENSIONS

A hierarchical system for audio classification and retrieval based on audio content analysis and modeling was presented in this paper. The audio recordings were first classified and segmented into speech, music, several types of

environmental sounds, and silence based on morphological and statistical properties of the temporal curves of three short-time features. This procedure is generic and model free, and achieved an accuracy rate of more than 90% tested with our audio database. In the next steps, sounds were further classified into finer classes within each basic type, and content-based audio retrieval was acomplished on top of the achiving scheme. We focused on modeling environmental sound with the hidden Markov model for the fine-level audio classification and audio retrieval. Two kinds of perceptual features of audio, i.e. timbre and rhythm, are included in the model by extracting features from the short-time spectrum of audio signals. We believe that timbre and rhythm together determine how a sound sounds to us. Preliminary experiments showed that accuracy rate of over 80% can be achieved with the proposed fine classification method. Results of audio retrieval also proved the HMM-based approach to be promising.

Future work will be done to refine the proposed system. First, we would like to enhance the coarse-level classification by taking hybrid-type sound and sound with noise into consideration. Second, we will look for more efficient feature vectors in the fine-level classification. Third, we want to investigate better ways in fixing model-matching mode and normalizing likelihood values.

8 REFERENCES

[1] J. Saunders: "Real-Time Discrimination of Broadcast Speech/Music", *Proc. ICASSP'96*, vol.II, pp.993-996, Atlanta, May, 1996

[2] E. Scheirer, M. Slaney: "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator", *Proc. ICASSP'97*, Munich, Germany, April, 1997

[3] L. Wyse, S. Smoliar: "Toward Content-based Audio Indexing and Retrieval and a New Speaker Discrimination Technique", downloaded from *http://www.iss.nus.sg/People/lwyse/lwyse.html*, Institute of Systems Science, National Univ. of Singapore, Dec., 1995

[4] D. Kimber, L. Wilcox: "Acoustic Segmentation for Audio Browsers", *Proc. Interface Conference*, Sydney, Australia, July, 1996

[5] A. Ghias, J. Logan, D. Chamberlin: "Query By Humming - Musical Information Retrieval in An Audio Database", *Proc. ACM Multimedia Conference*, pp.231-235, Anaheim, CA, 1995

[6] J. Foote: "Content-Based Retrieval of Music and Audio", *Proc. SPIE'97*, Dallas, 1997

[7] E. Wold, T. Blum, D. Keislar, *et al.*: "Content-Based Classification, Search, and Retrieval of Audio", *IEEE Multimedia*, pp.27-36, Fall, 1996

[8] Z. Liu, J. Huang, Y. Wang, *et al.*: "Audio Feature Extraction and Analysis for Scene Classification", *Proc. of IEEE 1st Multimedia Workshop*, 1997

[9] N. Patel, I. Sethi: "Audio Characterization for Video Indexing", *Proc. SPIE on Storage and Retrieval for Still Image and Video Databases*, Vol.2670, pp.373-384, San Jose, 1996

[10] T. Zhang, C.-C. Kuo: "Content-based Classification and Retrieval of Audio", *SPIE's 43rd Annual Meeting - Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations VIII,* San Diego, July 1998

[11] E. Miyasaka: "Timbre of Complex Tone Bursts with Time Varying Spectral Envelope", *Proceedings of ICASSP'82*, Vol.3, pp.1462-5, Paris, May 1982

[12] F. Everest: *The Master Handbook of Acoustics*, McGraw-Hill, Inc., 1994

[13] D. Reynolds, R. Rose: "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, Vol.3, No.1, pp.72-83, 1995

[14] L. Rabinar, B. Juang: *Fundamentals of Speech Recognition*, Prentice-Hall, Inc., New Jersey, 1993

[15] S. Bow: *Pattern Recognition*, Marcel Dekker, Inc., 1984