Content Analysis for Audio Classification and Segmentation

Lie Lu, Hong-Jiang Zhang, *Senior Member, IEEE*, and Hao Jiang

***Abstract*—In this paper, we present our study of audio content analysis for classification and segmentation, in which an audio stream is segmented according to audio type or speaker identity. We propose a robust approach that is capable of classifying and segmenting an audio stream into speech, music, environment sound, and silence. Audio classification is processed in two steps, which makes it suitable for different applications. The first step of the classification is speech and nonspeech discrimination. In this step, a novel algorithm based on K-nearest-neighbor (KNN) and linear spectral pairs-vector quantization (LSP-VQ) is developed. The second step further divides nonspeech class into music, environment sounds, and silence with a rule-based classification scheme. A set of new features such as the noise frame ratio and band periodicity are introduced and discussed in detail. We also develop an unsupervised speaker segmentation algorithm using a novel scheme based on quasi-GMM and LSP correlation analysis. Without a priori knowledge, this algorithm can support the open-set speaker, online speaker modeling and real time segmentation. Experimental results indicate that the proposed algorithms can produce very satisfactory results.**

***Index Terms*—Audio classification and segmentation, audio content analysis, speaker change detection, speaker segmentation.**

I. Introduction

AUDIO classification and segmentation can provide useful information for both audio content understanding and video content analysis. Therefore, in addition to the classical works on audio content analysis for audio classification and audio retrieval [1], [2], recent works have also integrated audio and visual information [12]–[14] in video structure parsing and content analysis. In general, the application of audio content analysis in video parsing can be considered in two parts. One is to classify or segment an audio stream into different sound classes such as speech, music, environment sound, and silence; the other is to classify speech streams into segments of different speakers. In this paper, our research works on these two tasks will be presented.

Intensive studies have been conducted on audio classification and segmentation by employing different features and methods. In spite of these research efforts, high-accuracy audio classification is only achieved for simple cases such as speech/music discrimination. Pfeiffer *et al.* [3] presented a theoretical framework and application of automatic audio content analysis using some perceptual features. Saunders [4] presented a speech/music classifier based on simple features such as zero-crossing rate and short-time energy for radio broadcast. The paper reported that the accuracy rate can achieve 98% when a window size of 2.4 s is used. Meanwhile, Scheirer *et al.* [5] introduced more features for audio classification and performed experiments with different classification models including GMM, BP-ANN, and KNN. When using a window of the same size (2.4 s), the reported error rate is 1.4%. However, it is found that these simple features-based methods cannot offer satisfactory results particularly when a smaller window is used or when more audio classes such as environment sounds are taken into consideration.

Many other works have been conducted to enhance audio classification algorithms. In [6], audio recordings are classified into speech, silence, laughter, and nonspeech sounds, to segment discussion recordings in meetings. In the work by Zhang and Kuo [7], pitch tracking methods were introduced to discriminate audio recordings into classes such as songs and speeches over music, based on a heuristic-based model. Accuracy of greater than 90% was reported. Srinivasan [12] proposed an approach to detect and classify audio that consists of mixed classes such as combinations of speech and music together with environment sound. The accuracy of classification is more than 80%.

In this paper, we present a high-accuracy algorithm for audio classification and segmentation. Speech, music, environment sound, and silence, the basic sets required in audio/video content analysis, are discriminated in a 1-s window, which is shorter than the testing unit used in [4] and [5]. Compared to other methods, our algorithm is computationally inexpensive and more practical for different applications. In order to improve the classification of the four audio classes in term of accuracy and robustness, a set of new features including *band periodicity* is proposed and discussed in detail.

Another novel work contributed in this paper is the real-time unsupervised speaker segmentation. We segment a speech sequence into segments of different speakers. Unlike general speaker identification or verification, no prior knowledge about the number and identities of speakers in an audio clip are assumed. In video browsing, if a speaker is first registered, a traditional speaker identification algorithm can be used, just as in the work of Brummer [16]. In video parsing applications, the knowledge of speakers is often not available or difficult to acquire. Therefore, it is desirable to perform unsupervised speaker segmentation in audio analysis.

There are several reported works on unsupervised speaker identification and clustering. Sugiyama [17] studied a simpler case, in which the number of the speakers to be clustered was

Manuscript received December 5, 2001; revised July 11, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C.-C. Jay Kuo.

L. Lu and H.-J. Zhang are with Microsoft Research Asia, Beijing 100080, China (e-mail: llu@microsoft.com; hjzhang@microsoft.com).

H. Jiang was with Microsoft Research Asia, Beijing 100080, China. He is now with Simon Fraser University, Vancouver, BC V6B 5K3, Canada (e-mail: hjiangb@cs.sfu.ca).

Digital Object Identifier 10.1109/TSA.2002.804546

assumed known. Wilcox [18], in contrast, proposed an algorithm based on HMM segmentation, where an agglomerative clustering method is used when the prior knowledge of speakers is unknown. Another system [19], [20] was proposed to separate controller speech and pilot speech with the GMM model, in addition to the speech and noise detection that were also considered in the framework. Speaker discrimination from the telephone speeches was studied in [21] using HMM segmentation. However, in this system, the number of speakers was limited to two. Mori [22] addresses the problem of speaker changes detection and speaker clustering without *a priori* speaker information available. Chen [23] also presented an approach to detect changes in speaker identity, environmental and channel conditions by using the Bayesian information criterion. An accuracy of 80% was reported.

Previous efforts to tackle the problem of unsupervised speaker clustering consist of clustering audio segments into homogeneous clusters according to speaker identity, background conditions, or channel conditions. Most methods used can be classified into two categories. One is based on VQ or clustering (GMM model), the other one is based on HMM model. A deficiency of these models is that they cannot meet the real-time requirement, since a computationally intensive iterative operation is utilized.

Real-time speaker segmentation is required in many applications, such as speaker tracking in real-time news-video segmentation and classification, or real-time speaker adapted speech recognition. In this paper, we present a real-time, yet effective and robust speaker segmentation algorithm based on LSP correlation analysis. Both the speaker identities and speaker number are assumed unknown. The proposed incremental speaker updating and segmental clustering schemes ensure our method can be processed in real-time with limited delay.

Fig. 1 shows the basic processing flow of the proposed approach that integrates audio segmentation and speaker segmentation. After feature extraction, the input digital audio stream is classified into speech and nonspeech. Nonspeech segments are further classified into music, environmental sound, and silence, while speech segments are further segmented by speaker identity. Detail processing will be discussed in the remaining sections.

The rest of the paper is organized as follows. Section II discusses in detail the audio features used in audio segmentation and speaker segmentation. Section III presents the audio classification and segmentation scheme. In Section IV, speaker segmentation algorithm is proposed. In Section V, empirical experiments and performance evaluation of the proposed algorithms are presented.

II. Feature Analysis

In order to improve the accuracy of classification and segmentation for audio sequence, it is critical to select good features that can capture the temporal and spectral characteristics of audio signal or the characteristics of speaker vocal tract. We select following features to classify or segment audio stream, *high zero-crossing rate ratio* (*HZCRR*), *low short-time energy ratio* (*LSTER*), *spectrum flux* (*SF*), *LSP divergence distance*, *band periodicity* (*BP*), and *noise frame ratio* (*NFR*). *LSP*, based on the work presented in [8], is also employed in our unsupervised speaker segmentation algorithm. These features will be described in detail in this section.

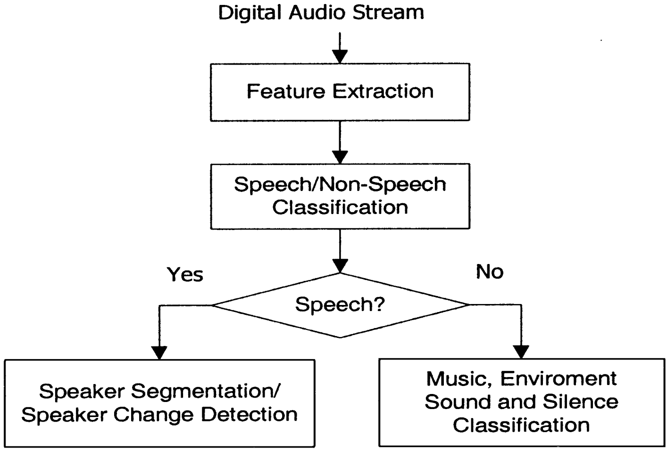

Digital Audio Stream
Feature Extraction
Speech/Non-Speech Classification
Yes
No
Speech?
Speaker Segmentation/ Speaker Change Detection
Music, Enviroment Sound and Silence Classification

Fig. 1. Basic processing flow of audio content analysis.

A. High Zero-Crossing Rate Ratio

Zero-crossing rate (*ZCR*) is proved to be useful in characterizing different audio signals. It has been popularly used in speech/music classification algorithms. In our experiments, we have found that the variation of *ZCR* is more discriminative than the exact value of *ZCR*. Therefore, we use *high zero-crossing rate ratio* (*HZCRR*) as one feature in our approach.

HZCRR is defined as the ratio of the number of frames whose *ZCR* are above 1.5-fold average zero-crossing rate in an 1-s window, as

$$HZCRR = \frac{1}{2N}\sum_{n=0}^{N-1}[\mathrm{sgn}(ZCR(n) - 1.5\ avZCR) + 1] \quad (1)$$

where n is the frame index, *ZCR*(n) is the zero-crossing rate at the nth frame, N is the total number of frames, *avZCR* is the average *ZCR* in a 1-s window; and sgn[.] is a sign function, respectively.

In general, speech signals are composed of alternating voiced sounds and unvoiced sounds in the syllable rate, while music signals do not have this kind of structure. Hence, for speech signal, its variation of zero-crossing rates (or *HZCRR*) will be in general greater than that of music, as shown in Fig. 2.

Fig. 2 illustrates the probability distribution curves of *HZCRR* for speech and music signals. The curves are obtained from our audio database using 1-s windows. It can be seen that the center of *HZCRR* distribution of speech segment is around 0.15, while *HZCRR* values of music segments mostly fall below 0.1, though there are significant overlaps between these two curves. Suppose we only use *HZCRR* to discriminate speech from music and use the cross-point of two curves in Fig. 2 as a threshold, the discrimination error rate would be 19.36%.

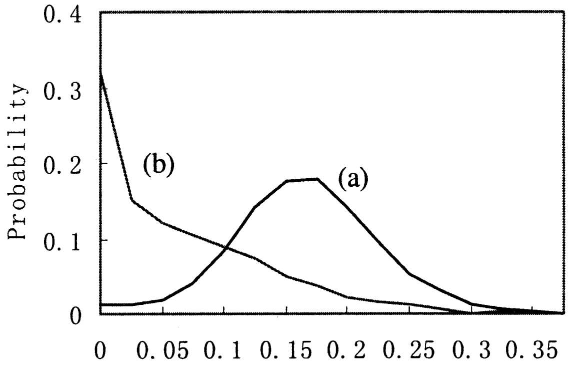


Fig. 2. Probability distribution curves of *HZCRR*. (a) Speech and (b) music.

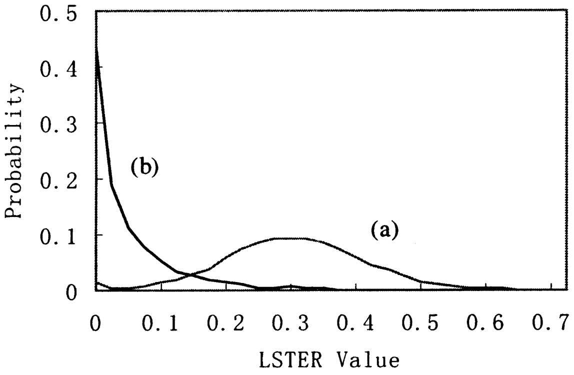


Fig. 3. Probability distribution curves of *LSTER*. (a) Speech and (b) music.

B. Low Short-Time Energy Ratio

Similar to *ZCR*, we also selected the variation, instead of the exact value, of short-time energy as one component of our feature vector. Here, we use *low short-time energy ratio* (*LSTER*) to represent the variation of short-time energy (*STE*).

LSTER is defined as the ratio of the number of frames whose *STE* are less than 0.5 time of average short-time energy in a 1-s window, as the following:

$$LSTER = \frac{1}{2N}\sum_{n=0}^{N-1}[\mathrm{sgn}(0.5\ avSTE\text{–}STE(n)) + 1] \quad (2)$$

where N is the total number of frames, *STE*(n) is the short-time energy at the nth frame, and *avSTE* is the average *STE* in a 1-s window.

LSTER is an effective feature, especially for discriminating speech and music signals. In general, there are more silence frames in speech than in music; as a result, the *LSTER* measure of speech will be much higher than that of music. This can be seen clearly from the probability distribution curves of *LSTER* for speech and music signals, as illustrated in the Fig. 3. It is shown that *LSTER* value of speech is around 0.15 to 0.5, while that of music is mostly less than 0.15. Based on Fig. 3, if we use the cross-point of two *LSTER* curves as a threshold to discriminate speech from music, the error rate would be only 8.27%. Therefore, *LSTER* is a good discriminator between speech and music.

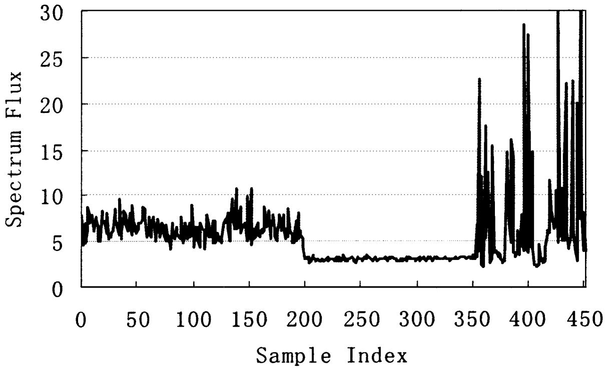


Fig. 4. Spectrum flux curve (0–200 s is speech, 201–350 s is music, and 351–450 s is environment sound).

C. Spectrum Flux

Spectrum flux (*SF*) is defined as the average variation value of spectrum between the adjacent two frames in a 1-s window

$$SF = \frac{1}{(N-1)(K-1)}\sum_{n=1}^{N-1}\sum_{k=1}^{K-1} \cdot[\log(A(n,k)+\delta) - \log(A(n-1,k)+\delta)]^2 \quad (3.1)$$

where $A(n, k)$ is the discrete Fourier transform of the nth frame of input signal

$$A(n,k) = \left|\sum_{m=-\infty}^{\infty} x(m)w(nL-m)e^{-j(2\pi/L)\,km}\right| \quad (3.2)$$

and $x(m)$ is the original audio data, $w(m)$ is the window function, L is the window length, K is the order of DFT, N is the total number of frames and δ is a very small value to avoid calculation overflow.

In our experiments, we found that, in general, the *SF* values of speech are higher than those of music. In addition, the environment sound is among the highest and changes more dramatically than the other two types of signals. Fig. 4 shows an example of spectrum flux of speech, music and environment sound. The speech segment is from 0 to 200 s, the music segment is from 201 to 350 s and the environment sound is from 351 to 450 s. Therefore, *SF* is a good feature to discriminate speech, environmental sound and music. This feature will be used in both speech/nonspeech classification and music/environment sound classification.

D. Band Periodicity

Band periodicity (*BP*) is defined as the periodicity of a subband. It can be derived by subband correlation analysis. Here, we have chosen four subbands: 500~1000 Hz, 1000~2000 Hz, 2000~3000 Hz, and 3000~4000 Hz, respectively. The periodicity property of each subband is represented by the maximum local peak of the normalized correlation function. For example, for a sine wave, its *BP* is 1; but for white noise, its *BP* is 0.

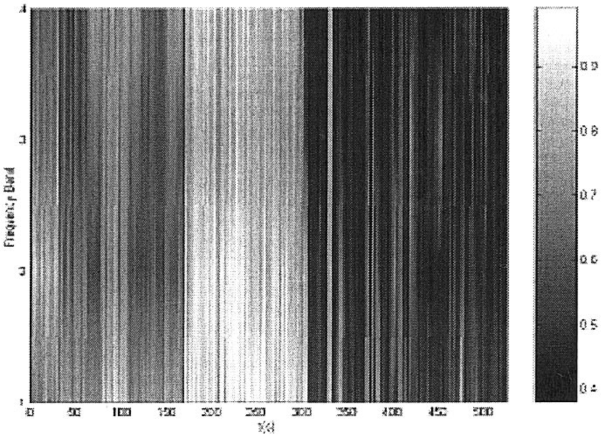


Fig. 5. Band periodicity of an example audio segment.

The normalized correlation function is calculated from the current frame and previous frame

$$r_{i,j}(k) = \frac{\sum\limits_{m=0}^{M-1} s(m-k)s(m)}{\sqrt{\sum\limits_{m=0}^{M-1} s^2(m-k)}\sqrt{\sum\limits_{m=0}^{M-1} s^2(m)}} \tag{4.1}$$

where $r_{i,j}(k)$ is the normalized correlation function; i is the band index, and j is the frame index. $s(n)$ is the subband digital signal of current frame and previous frame, when $n >= 0$, the data is from the current frame; otherwise, the data is obtained from the previous frame. M is the total length of a frame.

We denote the maximum local peak as $r_{i,j}(k_p)$, where k_p is the index of the maximum local peak, i is the band index and j is the frame index. That is, $r_{i,j}(k_p)$ is band periodicity of the ith sub-band of the jth frame. Thus, the band periodicity is calculated as

$$bp_i = \frac{1}{N}\sum_{j=1}^{N} r_{i,j}(k_p) \qquad i = 1, \ldots 4 \tag{4.2}$$

where bp_i is the band periodicity of ith sub-band, N is the total frame number in one audio clip.

Fig. 5 shows an example of band periodicity comparison between music and environment sounds. The music segment in the example is from 0 to 300 s, while the remaining part is environment sounds. The vertical axis represents different frequency sub-bands. It is observed that the band periodicities of music are in general much higher than those of environment sound. This is because music is more harmonic while environment sound is more random. Therefore, *band periodicity* is an effective feature in music/environment sound discrimination.

To show clearly the discrimination power of this feature, a probability distribution curve of *band periodicity* in the first sub-band for environment sound and music is illustrated in the Fig. 6. From Fig. 6, it can be obviously seen that the center of bp_1 value of environment sound is around 0.5, while bp_1 value of music is around 0.8, there is a considerable difference between them.

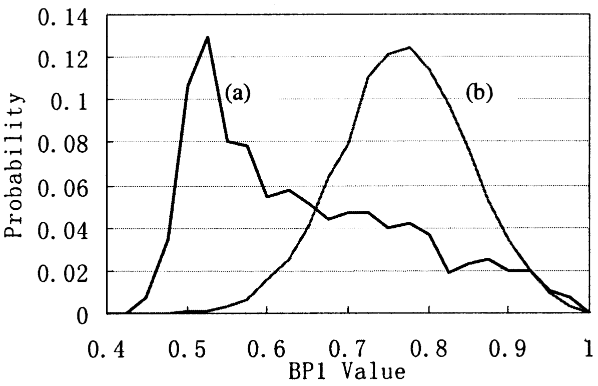


Fig. 6. Probability distribution curves of *BP1*. (a) Background sound and (b) music.

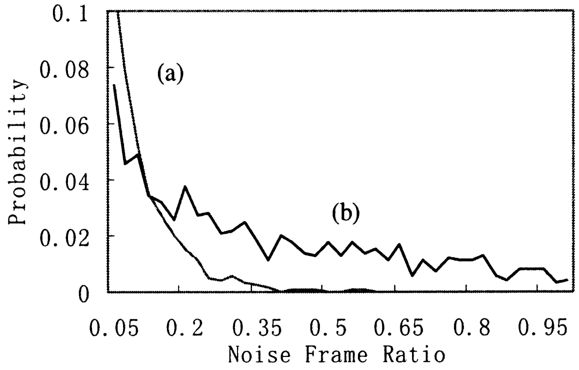


Fig. 7. Probability distribution curves of *NFR*. (a) Music and (b) environment sound.

In our implementation, only the periodicity of the first two bands bp_1 and bp_2 and the sum of the four bands' periodicity, *bpSum*, are used to discriminate music and environment sound.

E. Noise Frame Ratio

Noise frame ratio (*NFR*) is defined as the ratio of noise frames in a given audio clip. A frame is considered as a noise frame if the maximum local peak of its normalized correlation function is lower than a preset threshold. The *NFR* value of noise-like environment sound is higher than that of music, as illustrated in Fig. 7.

Fig. 7 shows the probability distribution curves of *NFR* for music and environment sounds from our audio database. For music, almost no *NFR* value is above 0.3; however, for environment sound, the portion of *NFR* values that are higher than 0.3 is much higher. *NFR* really depends on how noisy the signal is. Data shows some environment sound is more noise-like.

F. LSP Distance Measure

Linear spectral pairs (*LSPs*) are derived from linear predictive coefficients (LPC). Previous researches have shown that *LSP* has explicit difference in each audio class [10]. It is also found that *LSP* is more robust in the noisy environment [14].

K–L distance is used here to measure the *LSP* dissimilarity between two 1-s audio clips [8]

$$\begin{aligned} D = {} & \tfrac{1}{2}\, tr\left[(\boldsymbol{C}_{LSP} - \boldsymbol{C}_{SP})\left(\boldsymbol{C}_{SP}^{-1} - \boldsymbol{C}_{LSP}^{-1}\right)\right] \\ & + \tfrac{1}{2}\, tr\left[\left(\boldsymbol{C}_{SP}^{-1} + \boldsymbol{C}_{LSP}^{-1}\right)(\boldsymbol{u}_{LSP} - \boldsymbol{u}_{SP})(\boldsymbol{u}_{LSP} - \boldsymbol{u}_{SP})^T\right] \end{aligned} \tag{5}$$

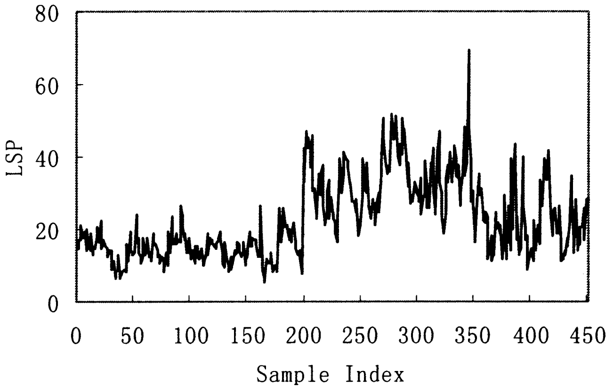


Fig. 8. LSP curve (0–200 s is speech; 201–350 s is music and 351–450 s is noisy speech).

where C_{LSP} and C_{SP} are the estimated *LSP* covariance matrices, $\boldsymbol{u}_{LSP}$ and $\boldsymbol{u}_{SP}$ are the estimated mean vectors, from two audio clip, respectively. In real implementation, ten-order *LSP* is extracted from each frame and then the covariance and mean are estimated from each audio clip.

The distance is composed of two parts. The first part is determined by the covariance of two segments and the second is determined by covariance and mean. Because the mean is easily biased by different environment condition, the second part is not considered and only the first part is used to represent the distance, similar to the work [8]. It is also similar to the cepstral mean subtraction (*CMS*) method used in speaker recognition to compensate the effect of environment conditions and transmission channels. Here, it is called divergence shape distance, which is defined by

$$D = \frac{1}{2}\, tr\left[(C_{LSP} - C_{SP})\left(C_{SP}^{-1} - C_{LSP}^{-1}\right)\right]. \qquad (6)$$

This dissimilarity measure is effective to discriminate speech and noisy speech from music. Fig. 8 shows an example of *LSP* distance between audio data and speech model obtained from our training data. The speech segment is from 0 to 200 s, the music segment is from 201 to 350 s and the noisy-speech segment is from 351 to 450 s. Obviously, the *LSP* distance is different among these classes. The distance between speech data and speech model is the smallest; while the distance between music and speech model is the largest.

LSP divergence shape is also a good measure to discriminate between different speakers. In our algorithm, we will use this feature to detect potential speaker change points with a 1-s step and a 3-s window.

Denote that the *LSP* covariance for pth s and qth s speech clip is C_p and C_q, respectively. According to (6), the dissimilarity measure of speaker models between these two speech clips can be defined as

$$D(p,\, q) = tr[(C_p - C_q)(C_q^{-1} - C_p^{-1})]. \qquad (7)$$

In general, if the dissimilarity is larger than a threshold, these two speech clips could be considered from two different speakers. Though it is a simple scheme, but it is capable of measuring the difference between speakers. An example of *LSP* distance between different speakers is illustrated in Fig. 9.

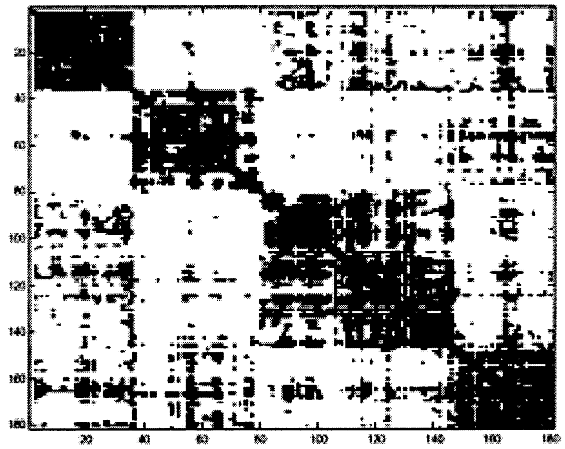


Fig. 9. LSP divergence distance map.

Fig. 9 shows the dissimilarities between any two 3-s speech subsegments in a 180-s-long speech. One threshold is used to transform $D(p,\, q)$ to binary value (0, 1). Value 0 is represented by black pixel, while value 1 is represented by white pixel. It can be clearly seen that the figure is symmetric, and there are four speakers in this speech segment.

III. Audio Classification and Segmentation

With the features presented in the previous section, a two-step scheme is proposed to classify audio clips into one of the four audio classes: speech, music, environment sound and silence. At the first step, an input audio stream is classified into speech and nonspeech segments by a K-nearest-neighbor (*KNN*) classifier and linear spectral pairs-vector quantization (*LSP-VQ*) analysis. In the second step, nonspeech segments are further classified into music, environmental sound, and silence by a rule-based scheme. This two-step scheme is suitable for different applications and is capable of achieving high classification accuracy. Based on these classification results, the segmentation of an audio stream is achieved. Postprocessing scheme is then applied to further reduce misclassification. The detailed system block diagram of the proposed audio classification and segmentation scheme is shown in Fig. 10.

In extracting audio features for our classification scheme, all input signals are downsampled into 8-KHz sample rate and subsequently segmented into subsegments by 1-s window. This 1-s audio clip is taken as the basic classification unit in our algorithms. If there are two audio types in 1-s audio clip, it will be classified as the dominant audio type. The audio clip is further divided into forty 25 ms nonoverlapping frames, on which a 15 Hz bandwidth expansion is applied. A feature vector is extracted based on these 40 frames in 1-s audio clip to represent the window. We use those features presented in the previous section to represent the characteristics of each 1-s audio clip.

A. Speech/Nonspeech Discrimination

The first step of our audio classification scheme is to discriminate speech and nonspeech segments. In this scheme, we first apply a *KNN* classifier based on *high zero-crossing rate ratio*

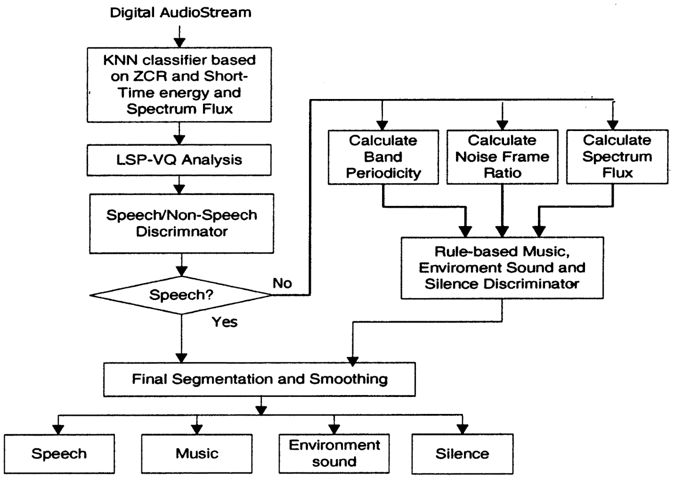

Digital AudioStream
KNN classifier based on ZCR and Short-Time energy and Spectrum Flux
LSP-VQ Analysis
Speech/Non-Speech Discrimnator
Speech?
No
Yes
Calculate Band Periodicity
Calculate Noise Frame Ratio
Calculate Spectrum Flux
Rule-based Music, Enviroment Sound and Silence Discriminator
Final Segmentation and Smoothing
Speech
Music
Environment sound
Silence

Fig. 10. Audio classification and segmentation system diagram.

(*HZCRR*), *low short-time energy ratio* (*LSTER*), and *spectrum flux* (*SF*) to perform a fast preclassification of speech and nonspeech. Then, we propose a refine scheme based on *LSP* analysis [8] to refine the classification results and make the final decision. Empirical experiments indicate that this scheme can get higher accuracy than just combining every feature.

1) Preclassification: Due to the discrimination power and low computational cost, we use *high zero-crossing rate ratio, low short-time energy ratio,* and *spectrum flux* to form a feature vector, $\{HZCRR, LSTER, SF\}$, for fast preclassification. However, since none of the three features can achieve a 100% accurate classification, a more sophisticated classification scheme based on *KNN* classifier and VQ analysis is proposed.

Suppose the generated feature vectors satisfy Gaussian mixture model, we can construct a number of speech codebooks and nonspeech codebooks by our training database. The training data for codebook generation is composed of four audio sequences of about 2 h from MPEG-7 test set CD1 and the other 100 environment sound clips of each about 4 s long. A *KNN*-2 classifier is used in our scheme to perform audio preclassification.

This preclassification scheme works well in most cases and is fast because of its computational simplicity. An exception is when the scheme is applied to signals of mixed audio types. As discussed in Section II, *HZCRR*, *LSTER*, and *SF* characterize the fluctuations of zero-crossing rate, short-time energy, and spectrum. However, these features of noisy speech signals are similar to those of music. Furthermore, these features of some music signals with the drum sounds as well as some environment sounds are often similar to those of speech signals. Since preclassifier alone can not assure high classification accuracy, we proposed a refining scheme to solve the problem due to mixed audio signals.

2) Refining Scheme: As presented in Section II, *LSP* is a robust feature in the noisy environment for effective discrimination between noisy speech and music, though it is relatively more computationally complex. Therefore, this feature is utilized to refine the preclassification results. In our scheme, we obtain a speech *LSP* covariance matrix model as a speech codebook, through a training process. The distance between the speech codebook and the *LSP* covariance of the testing audio clip is then compared. If the distance is smaller than a threshold, the audio clip is classified as speech; otherwise, it is classified as nonspeech.

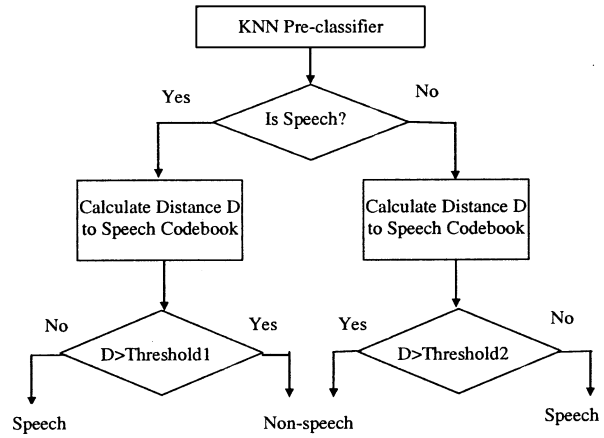


Fig. 11. Final speech/nonspeech discrimination.

The procedures for final classification of speech and nonspeech are illustrated in Fig. 11.

As shown in Fig. 11, the result of preclassification is examined by measuring the distance of an audio clip from the speech model codebook. We denoted the distance as D. Depending on the preclassification result, two thresholds are used in making the final decision. If the preclassification result is speech, then D is compared against *Threshold*1. If D is greater than *Threshold*1, the audio clip is classified as nonspeech. Otherwise, D is compared against *Threshold*2, and the same rule is applied to make final decision.

Here is some guide to setting *Threshold*1 and *Threshold*2. Supposing the *LSP* distance between speech clip and speech codebook satisfies a Gaussian distribution: $N(\mu_p, \sigma_p)$, almost no value (about 0.03%) will be larger than $\mu_p + 3\sigma_p$. If the distance of one clip is larger than this value, the clip is most likely is a nonspeech. Thus the *Threshold*1 can be set as

$$Threshold1 = u_p + \lambda_1 \cdot \sigma_p \tag{8.1}$$

where λ_1 is a coefficient that is usually set to be three.

Similarly, supposing the *LSP* distance between nonspeech clip and speech codebook satisfies Gaussian distribution $N(\mu_n, \sigma_n)$, then the *Threshold*2 can be set as

$$Threshold2 = u_n - \lambda_2 \cdot \sigma_n \tag{8.2}$$

where λ_2 is another coefficient that also usually set to be three.

In general, *Threshold*1 is greater than *Threshold*2. Thus, we can prevent too many preclassification results from being converted incorrectly.

In practical applications, four speech model codebooks are generated from training data using the Linde–Buzo–Gray (LBG) algorithm [11]. The training data include speeches by different speakers at different ages and of different genders, in various recording conditions. The dissimilarity of a test audio clip is defined as the minimum distance between the clip and the four speech model codebooks.

B. Music, Environment Sound, and Silence Classification Scheme

Nonspeech is further classified into music, environment sound, and silence segments. In our scheme, silence detection is performed first. Then, for nonsilence segment, it is classified into music or environment sound by applying a set of rules.

1) Detecting Silence: Silence detection is performed based on short-time energy and zero-crossing rate in 1-s windows. If the average short-time energy and zero-crossing rate is lower than a threshold, the segment is classified as silence; otherwise, it is classified as nonsilence segment. This simple scheme works well in our applications.

2) Discriminating Music From Environment Sound: Band periodicity (*BP*), *spectrum flux* (*SF*), and *noise frame ratio* (*NFR*) are used to discriminate music from environment sounds. *BP* acts as the basic measure. As shown in Fig. 5, the band periodicity of music is greater than that of environment sounds in most cases. However, it is noted that there are certain degree of overlaps in this feature distribution, which can lead to potential classification errors. To avoid this problem, *SF* and *NFR* are also used. From Fig. 4, the *SF* of environment sound is much higher than that of music in most of cases, while in Fig. 7, there is almost no *NFR* value of music higher than 0.35. Hence, these facts are utilized in our algorithm according to the following rule.

First, if any of the bp_1, bp_2, or *bpSum* of an audio clip is lower than the predefined thresholds, the clip is considered as a segment of environment sounds. Otherwise, it goes to next step.

Then, if *NFR* of a clip is greater than a given threshold, the clip is classified as noise-like environment sound. Otherwise it goes to third step, in which *SF* of the window is examined. If the *SF* is greater than a threshold, a clip is also classified as environment sound. This rule is useful especially for some strong periodicity environment sounds such as tone signal whose *BP* and *NFR* are similar to that of music signals. Only *spectrum flux* can distinguish them.

Finally, music segments can be segmented by excluding above conditions. This is because the *BP* values of for music signals are usually higher, but *NFR* and *SF* values are lower, compared to environmental sound.

The decision process is illustrated in Fig. 12, where Th_{BP}, Th_{NFR}, and Th_{SF} are thresholds for the feature *BP*, *NFR*, and *SF*, respectively.

The optimal thresholds could be obtained by searching the whole feature space in order to minimize the global misclassification. However, it will be very time-consuming. In fact, the thresholds could be constrained in a certain region when searching the optimal one according to the feature distribution characteristics. Suppose the mean and covariance of each feature of music are (μ_{BP}, σ_{BP}), $(\mu_{NFR}, \sigma_{NFR})$, and (μ_{SF}, σ_{SF}), respectively, the constraining regions are set as

$$Th_{BP} \in [\mu_{BP} - 3\sigma_{BP}, \mu_{BP}] \tag{9.1}$$

$$Th_{NFR} \in [\mu_{NFR}, \mu_{NFR} + 3\sigma_{NFR}] \tag{9.2}$$

$$Th_{SF} \in [\mu_{SF}, \mu_{SF} + 3\sigma_{SF}]. \tag{9.3}$$

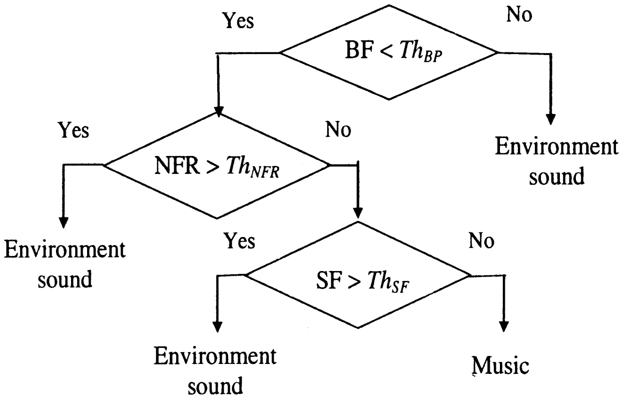


Fig. 12. Music/environment sound discrimination process.

Then, the optimal thresholds $(Th_{BP}, Th_{NFR}, Th_{SF})$ can be exhaustively searched in the above space. In actual implementation, we discretize each dimension into 40 values. For each value, we can get the misclassification result. The value which resulted in least error is taken as an optimal threshold.

C. Final Segmentation and Smoothing

Final segmentation of an audio stream is achieved by classifying each 1-s window into an audio class. Meanwhile, considering that the audio stream is always continuous in video program, it is highly impossible to change the audio types too suddenly or too frequently. Under this assumption, we apply smoothing rules in final segmentation of an audio sequence. The first rule used is

Rule 1 if $(s[1] \neq s[0] \&\& s[2] = s[0])$ then $s[1] = s[0]$

where a 3-s sequence is considered, $s[0]$, $s[1]$, $s[2]$ stands for the audio type of previous 2-s, previous second and current second, respectively. This rule implies that if the middle second is different from the other two while the other two are the same, the middle one is considered as misclassification. For example, if we detect a pattern of consecutive sequence like "speech–music–speech," it is most likely the sequence should belong to speeches. But for sequence such as "speech–music-environment sound," the middle second can either be correct or incorrect classification. We can also optionally rectify the middle second as the previous or the succeeding audio type. In our approach, we will uniformly rectify the middle second according to its previous audio type. That is

Rule 2 if $(s[1] \neq s[0] \&\& s[2] \neq s[0] \&\& s[2] \neq s[1])$
then $s[1] = s[0]$.

It should be noted that rules 1 and 2 are not applicable to silence second, since silence is highly possible to appear in 1-s window. That is, the sequence such as "speech–silence–speech" is accepted. Consequently, by combining the above three rules, the final rule becomes

Final Rule if $(s[1] \neq s[0] \&\& s[1] \neq SILENCE \&\& s[2]$
$\neq s[1])$ then $s[1] = s[0]$.

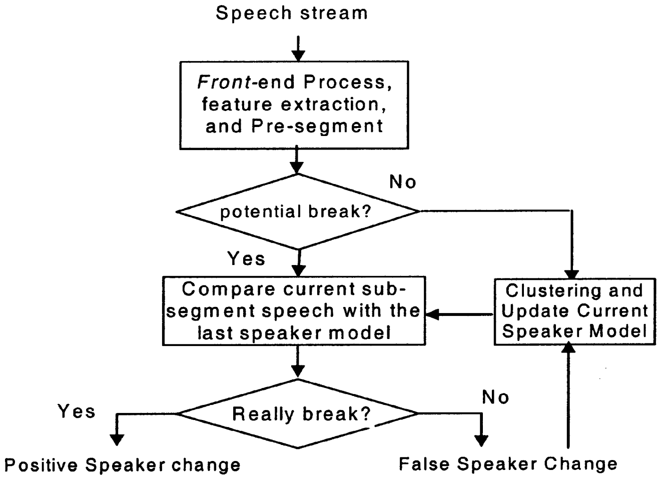

Speech stream
Front-end Process, feature extraction, and Pre-segment
potential break?
No
Yes
Compare current sub-segment speech with the last speaker model
Clustering and Update Current Speaker Model
Really break?
Yes
No
Positive Speaker change
False Speaker Change

Fig. 13. Flow diagram for speaker change detection.

IV. Speaker Segmentation

If an audio segment belongs to speech, we further segment it according to the speakers, i.e., the speaker transitions are detected in the speech segments. As mentioned in Section II, *LSP* analysis will be applied to speaker changes detection. Fig. 13 illustrates our unsupervised speaker change detection algorithm. The algorithm is mainly composed of three modules. They are front-end process module, segmentation module, and the module for clustering and updating speaker model.

The input speech stream is first segmented into 3-s subsegments with 2-s overlapping. That is, the step or temporal resolution of the segmentation is 1 s. Each subsegment is preprocessed by removing silence and unvoiced frame. Presegment is processed to find potential speaker change point. If the boundary of a potential speaker change is not detected, current speaker model is updated incrementally. Otherwise, Bayesian information criterion (*BIC*) is employed to verify the correctness of detected boundary.

A. Front-End Processing

The input audio stream is first downsampled into 8 KHZ, 16 bits, mono channel, and preemphasized, a common format as used in the audio classification process. The speech stream is then divided into small subsegments by a 3-s window with 2-s overlapping. The subsegment is further divided into nonoverlapping 25-ms-long frames. The most important feature extracted from each frame is *LSP* vector. Other extracted features include short-time energy (*STE*) and zero-crossing rate (*ZCR*). They are used to discriminate silence frames and unvoiced frames, which should be excluded when estimating speaker model.

B. Potential Speaker Change Detection

At this step, speaker model is extracted for each subsegment, *LSP* divergence distance is used to measure the dissimilarity between each two neighboring speaker models at each time slot, as shown in Fig. 14(a). Thus, if a local maximum is found in the *LSP* distance series, and furthermore if it is larger than a predefined threshold, it is taken as a potential speaker change point.

Let $D(i, j)$ denote the distance between the ith and jth speech subsegment, as defined by (8). A potential speaker change is detected between ith and $(i+1)$th speech subsegment, if the following conditions are satisfied:

$$\begin{aligned} &D(i, i+1) > D(i+1, i+2), \\ &D(i, i+1) > D(i-1, i), \qquad D(i, i+1) > Th_i \end{aligned} \tag{10}$$

where Th_i is a threshold.

The first two conditions guarantee a local peak exists, and the last condition can prevent very low peaks from being detected. Reasonable results can be achieved by using this simple criterion. However, the threshold is difficult to set *a priori*. If the threshold is too small, false detection would be easily generated. False detection could be reduced by increasing the threshold, with the expense that some positive speaker change boundaries could be missed. The threshold is affected by factors such as insufficient estimate data and different environment conditions. For example, from our experiments, we found that the distance between speech subsegments will increase if the speech is in a noisy environment. Therefore, the threshold should be increased accordingly in a noisy environment. To obtain a more robust threshold, an automatic threshold setting method is proposed as follows.

In our algorithm, the threshold is automatically set according to the previous N successive distances, i.e.,

$$Th_i = \alpha \cdot \frac{1}{N} \sum_{n=0}^{N} D(i-n-1, i-n) \tag{11}$$

where N is the number of previous distances used for predicting threshold, and α is a coefficient used as an amplifier. We set $\alpha = 1.2$ in our algorithm. The threshold determined in this way works satisfactory in different conditions. However, the false detections can still exist due to the insufficient data in estimating the speaker model accurately from only one short speech subsegment. The estimated speaker model would be biased in this case.

In order to solve this problem, we should use as much data as possible to update speaker model. A more accurate refinement method is proposed to refine the above potential speaker change boundaries.

C. Incremental Speaker Model Updating

In order to collect as much data as possible to estimate speaker model more accurately, we utilize the detection results of potential speaker change. If no potential speaker change point is detected, the next subsegment is assumed as the same speaker as the previous one. Thus, we update the current speaker model using this available new data, as shown in Fig. 14(b).

GMM-32 is used to model a speaker. The model is established progressively as more and more data become available. Initially, there is no sufficient speaker data, thus GMM-1 is used. When more speaker data are available, the model will grow up to GMM-32 gradually.

In general, EM algorithm is used to estimate the Gaussian mixture models. However, two problems will be introduced. First, it causes storage overhead since all feature data are required to be saved in memory or disk. Second, the recur-

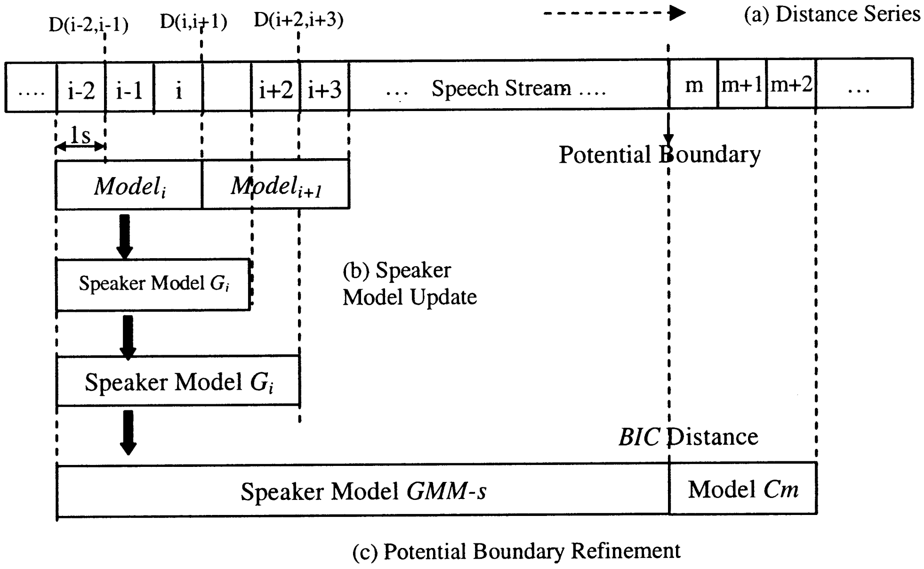


Fig. 14. Illustration of speaker change detection.

sive process of EM algorithm could not guarantee real-time processing. Therefore, we introduce an alternative clustering method which is less time consuming. Although the accuracy is not as high as the EM algorithm, it works well most of the time. The detail algorithm is described in the following.

Suppose the current speaker model $G_i \sim N(u, C)$ is obtained from the previous $(M-1)$ subsegments and there is no potential speaker change point between $(M-1)$th and Mth speech segment, it implies these segments belong to the same speaker. Thus, we update the current speaker model G_i using the feature data of the Mth segment. If the model of Mth speech segment is $N(u_m, C_m)$, the current speaker model could be updated as

$$C' = \frac{N}{N+N_m} C + \frac{N_m}{N+N_m} C_m + \frac{N \cdot N_m}{(N+N_m)^2} \cdot (\mu - \mu_m)(\mu - \mu_m)^T \quad (12)$$

where N and N_m is the number of feature vectors used for modeling $N(u, C)$ and $N(u_m, C_m)$, respectively.

The third part of (12) is determined by the means. However, the means can be easily biased by different environment conditions. In practice, we ignore the mean part of (12) to compensate the effect of different environment conditions and transmission channel. Then, (12) is simplified as

$$C' = \frac{N}{N+N_m} C + \frac{N_m}{N+N_m} C_m. \quad (13)$$

The above procedure is looped till the dissimilarity between the speaker models before and after updating is small enough or a potential speaker change point is met. The dissimilarity is also measured by the *LSP* divergence shape distance. When the dissimilarity is small enough, it is assumed that the current Gaussian model is estimated accurately with sufficient training data. In other words, it is not necessary to continue updating G_i. The next Gaussian model, G_{i+1}, is initiated and updated with the new data using the same method.

For one speaker, several Gaussian models will be estimated by the above method. This is called segmental clustering since each component is obtained from one speaker segment. Combining these Gaussian models would form a quasi-Gaussian mixture model. The weight of each Gaussian model is set by their corresponding number of training data. Supposing the quasi-Gaussian mixture model for a speaker is GMM-s, in which each Gaussian model G_i is estimated by N_i feature vectors ($i = 1, \ldots s$). Then, the weight w_i of the ith Gaussian model G_i is computed by

$$w_i = N_i / N \quad (14)$$

where $N = \sum_{i=1}^{S} N_i$ is the total number of feature vectors.

By using this method, the speaker model will grow from GMM-1, GMM-2, up to GMM32. When the GMM32 is reached, the updating of the speaker model is terminated. This method (quasi-GMM by segmental clustering) is slightly different from the original GMM. It tends to neglect low-weighted components in a GMM and is less accurate than GMM obtained using EM algorithms. Nevertheless, it still can capture the most important components in GMM, and furthermore, real-time requirement is met due to its computational simplicity. Through our empirical experiments, it could achieve reasonable accuracy.

D. Speaker Change Boundary Refinement

There are false positives in the potential speaker change points obtained with the algorithms described in Section IV-B. To remove false positives and detect only real speaker change boundaries, a refinement algorithm is used. The algorithm is based on the dissimilarity between the current segment and the previous speaker model obtained from the segments before the current potential boundary. In this step, Bayesian information criterion (BIC) [23], [24] is used to measure the dissimilarity, as shown in Fig. 14(c).

Suppose two Gaussian model from two speech clips are $N(u_1, C_1)$ and $N(u_2, C_2)$, the number of data used to estimate these two models are N_1 and N_2, respectively; and when one Gaussian Model is used to estimate these two speech clips, the model is $N(u, C)$. The BIC difference between the two models is

$$\begin{aligned} &BIC(C_1, C_2) \\ &\quad = \tfrac{1}{2}\left((N_1 + N_2)\log|C| - N_1 \log|C_1| - N_2 \log|C_2|\right) \\ &\qquad - \tfrac{1}{2}\,\lambda\left(d + \tfrac{1}{2}\,d(d+1)\right)\log(N_1 + N_2) \end{aligned} \tag{15}$$

where λ is a penalty factor to compensated for small size cases, and d is the feature dimension. Generally, $\lambda = 1$.

According to BIC theory, if $BIC(C_1, C_2)$ is positive, the two speech clips could be considered from different sources (speakers). The advantage of using BIC is that it is threshold free.

Suppose at the potential speaker boundary, the model of previous speaker is GMM-s, in which each Gaussian model is $N(u_i, C_i)$ $(i = 1, \ldots s)$; and the model of current segment is $N(u, C)$. Then the distance between them is estimated as the weighted sum of the distance between $N(u, C)$ and each $N(u_i, C_i)$

$$D = w_i \cdot \sum_{i=1}^{S} BIC(C_i, C). \tag{16}$$

This distance does not take the GMM-s as an integral one, but as several independent components. However, it is still reasonable since the GMM-s model is obtained from segmental clustering. That is, each component Gaussian model is obtained from an independent segment. The BIC distance considering one component of GMM-s can be used as the similarity confidence between the current segment and one segment of the previous speaker. Thus, the weighted sum (average distance) can be used to represent the distance between current segment and previous speaker.

Based on the aforementioned BIC theory, if $D > 0$, it must be a real speaker change boundary. If a candidate is not a real boundary, the speaker data is used to update the speaker model following the method previously described.

LSP divergence distance or Bayesian information criterion is not uniformly used at potential speaker boundary detection and refinement. The reason is as follows. At the step of potential speaker change detection, the data is too small to estimate a model accurately. Bayesian information criterion is found to be vulnerable by different words or different speakers, so false alarms can be easily generated. At the step of potential boundary refining, the model is more accurate; moreover, BIC could compensate different training data and is threshold free, while *LSP* divergence distance depends on thresholds. It is more efficient for BIC in this step, as shown in our experiments.

V. Experiment Results

A. Audio Classification and Segmentation Evaluations

The evaluation of the proposed audio classification and segmentation algorithms have been performed by using an audio database composing of data from MPEG-7 test data set CD1, TV news, movie clips, and some audio clips from the Internet. This database includes speech in various conditions, such as in record studios, speeches with telephone (4 kHz) bandwidth and 8 kHz bandwidth. The music content in this data set is mainly songs and pop music. Such music contents are difficult for most audio classifiers. The background sound in the database include many types, such as aviations, animals, autos, beeps, cartoon, combat, crowds, and so on. Two hours of data was used for training, and 4-h data was used for testing. The testing data includes about 9600 s speech, 3400 s music, and 1200 s environment sounds. The training data is approximately half of the testing data. In our experiments, we set 1 s as a test unit. If there are two audio types in a 1-s audio clip, we will classify it as the dominant audio type.

TABLE I
Speech, Music, Environment Sound Classification on Baseline System (Unit: 100%)

Sound Type	**Total Number**	**Discrimination Results**		
		Speech	**Music**	**ENV Sound**
Speech	100	95.46	2.81	1.73
Music	100	5.24	88.39	6.37
Environment Sound	100	15.25	22.87	61.88

TABLE II
Baseline Classification Result on Pure Speech and Noisy Speech (Unit: 100%)

Sound Type	**Total Number**	**Discrimination Results**	
		Speech	**Music**
Pure Speech	100	96.74	3.26
Noisy speech	100	73.62	26.38

TABLE III
Classification on Pure Speech and Noisy Speech After Refinement (Unit: 100%)

Sound Type	**Total Number**	**Discrimination Results**	
		Speech	**Music**
Pure Speech	100	98.23	1.77
Noisy speech	100	85.18	14.82

We first implement a baseline system which uses the feature (*HZCRR*, *LSTER*, *SF*) with clustering and the KNN method, as described in the Section III. The performance data are listed in Table I.

This baseline system works well for speech/nonspeech discrimination, but does not work well on environment sound. In our experiments, we also found that the baseline system has worse performance on noisy speech than pure speech. About 26.38% noisy speech is discriminated as music, as shown in the Table II. This is because some features of noisy speech are very similar to those of music, in particular the pop music.

These facts show that the base system is only effective as a preclassification process, and more improvements are expected. Therefore, we propose to use new features to increase the classification performance of noisy speech and environment sound. After the refinement scheme by *LSP* divergence shape, the performance is improved significantly, as shown in Table III.

After employing our music and environment classification scheme, the performance for environment classification is also

TABLE IV
SPEECH, MUSIC, ENVIRONMENT SOUND CLASSIFICATION BEFORE SMOOTHING (UNIT: 100%)

Sound Type	Total Number	Discrimination Results		
		Speech	Music	ENV Sound
Speech	100	96.73	1.89	1.38
Music	100	3.68	91.34	4.98
Environment Sound	100	11.49	9.24	79.27

TABLE V
FINAL RESULT OF SPEECH, MUSIC, ENVIRONMENT SOUND CLASSIFICATION (UNIT: 100%)

Sound Type	Total Number	Discrimination Results		
		Speech	Music	ENV Sound
Speech	100	97.45	1.55	1.00
Music	100	3.16	93.04	3.80
Environment Sound	100	10.49	5.08	84.43

TABLE VI
THE TOTAL ACCURACY RESULT FOR DIFFERENT DISCRIMINATION TYPE

Discrimination Type	Accuracy
Speech/music	98.03%
Speech/music/environment sound	96.51%

improved. The total performance of our system is showed in Table IV.

Considering the continuity of audio stream, a smoothing scheme is processed. The performance has been further improved as shown in Table V.

From Table V, we can see that speech, music, and environment sound can be well discriminated. 97.45% of speech samples are discriminated correctly; only 1.55% speech is mistakenly classified into music while 1.00% into environment sounds. The total accuracy of discriminating these three classes is as high as 96.51%. If only speech and music are considered, the accuracy reaches 98.03%. The final accuracy results of different discrimination types are listed in Table VI.

The experiments have shown that the proposed scheme achieves excellent classification accuracy.

B. Speaker Change Detection and Segmentation Evaluation

The testing materials used for speaker segmentation evaluations are news video programs from MPEG7 test data, CNN news, and CCTV news. In total, they are about 2 h. The audio track in the test set is sampled at 16 kHz, 32 kHz, or 44.1 kHz in one or two channels. In the experiments, each format audio is converted to 8 kHz and mono-channel before further processing.

Fig. 15 shows an example of 176-s-long speech. The speech segment includes four speaker change boundaries at 17 s, 52 s, 86 s, 154 s respectively. Fig. 15(a) shows the initial LSP distance between each two speech subsegments, the adaptive threshold and the potential speaker change boundaries. It can be seen that the number of potential boundaries are more than real boundaries. Fig. 15(b) shows the Bayesian information criterion at the potential speaker change boundary with speaker model updated by using as much data as possible. If the value is positive, it is considered as a real speaker change boundary. There are four boundaries could be detected from Fig. 15(b).

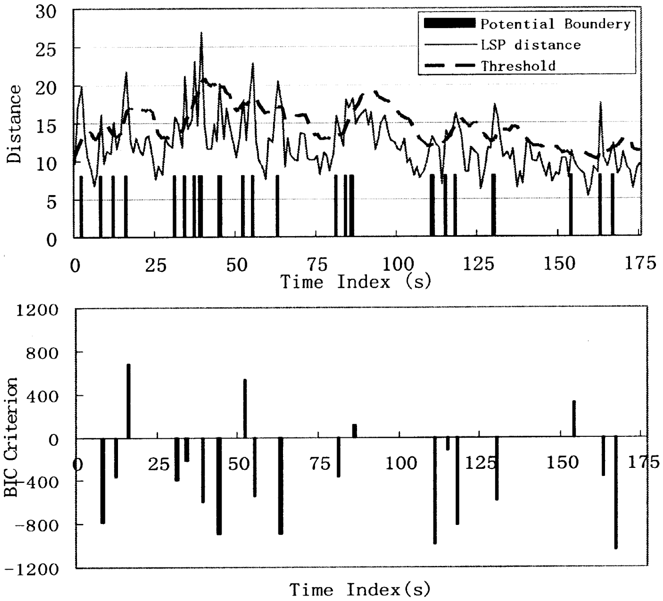


Fig. 15. Example of speaker change detection algorithm.

TABLE VII
SPEAKER CHANGE DETECTION ACCURACY

Video Clip	Original	Detected	Miss	False	Recall	Precision
1	5	5	0	0	100%	100%
2	32	38	0	6	100%	84.21%
3	29	32	3	6	89.46%	81.25%
4	37	38	6	7	83.78%	81.58%
5	27	29	3	5	88.89%	82.76%
6	41	41	6	6	85.37%	85.37%
7	17	19	1	3	94.12%	84.21%
All	188	202	19	23	89.89%	83.66%

The performance evaluations of speaker change detection are described with recall and precision. The results are listed in Table VII. Because false alarms are more tolerant than missed boundaries in the video content analysis, we assign higher cost to missed alarms. It can be noted from the table that the number of missed alarms is less than false alarms. The overall recall is 89.89% and the precision is 83.66%.

In the experiment, we have found that if there is a laugh burst between speeches, it is easily detected as speaker change boundary. This is because we have no more coming data to be used to compare with the previous one considering the real-time requirement with low delay. It is also found that the same speaker in different environment sometimes is easily detected as different ones. This indicates that our compensation for the effect of environment conditions and transmission channel is insufficient. The problem would remain a challenge in the speaker recognition field and may have a long way to go.

C. Computation Complexity

We have also tested the computational complexity of our algorithm in term of CPU time. With a Pentium III 667 MHz PC with Windows 2000, the whole process, including audio segmentation and speaker segmentation, can be completed in about 20% of the length of an audio/video clip. The correlation calculation in computing *LSP* matrix and band periodicity is the most time-consuming part in our algorithm. After using an optimized function to compute these features, the time performance has been increased dramatically. Therefore, our audio classification and speaker segmentation scheme is able to meet the real-time requirement in multimedia applications.

VI. Conclusions

In this paper, we have presented our study on audio classification and segmentation for applications in audio/video content analysis. We have described in detail a novel audio segmentation and classification scheme that segments and classifies an audio stream into speech, music, environment sound, and silence. These classes are the basic data set for audio/video content analysis. The algorithm has been developed and presented in two stages, which is very suitable for different applications. We also have introduced a set of new features, such as *noise frame ratio* and *band periodicity*, which have high discrimination power among different audio types. Experimental evaluation has shown that the proposed audio classification scheme is very effective and the total accuracy rate is over 96%. The novel scheme and new features introduced ensure that the system can achieve high accuracy even with a smaller testing unit.

We have also developed an improved approach on unsupervised speaker segmentation based on *LSP* divergence analysis. Incremental speaker modeling and adaptive threshold setting have been described in detail, which makes unsupervised speaker segmentation possible. Segmental clustering, which requires less computation, has also been proposed, so that the algorithm can totally suit the real-time processing in multimedia application. Experiments have shown that the algorithm is considerably effective. The overall recall is up to 89.89%, and the precision is 83.66%.

In the future, our audio classification scheme will be improved to discriminate more audio classes. We will improve the performance of our speaker segmentation algorithm and extend it to speaker tracking. We will also focus on developing an effective scheme to apply audio content analysis to assist video content analysis and indexing.

References

[1] J. Foote, "Content-based retrieval of music and audio," *Proc. SPIE*, vol. 3229, pp. 138–147, 1997.

[2] E. Wold, T. Blum, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, 1996.

[3] S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," in *Proc. 4th ACM Int. Conf. Multimedia*, 1996, pp. 21–30.

[4] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proc. ICASSP'96*, vol. II, Atlanta, GA, May 1996, pp. 993–996.

[5] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature music/speech discriminator," in *Proc. ICASSP' 97*, Apr. 1997, vol. II, pp. 1331–1334.

[6] D. Kimber and L. Wilcox, "Acoustic segmentation for audio browsers," in *Proc. Interface Conf.*, Sydney, Australia, July 1996.

[7] T. Zhang and C.-C. J. Kuo, "Video content parsing based on combined audio and visual information," *Proc. SPIE*, vol. IV, pp. 78–89, 1992.

[8] J. P. Campbell, Jr., "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[9] A. V. McCree and T. P. Barnwell, "Mixed excitation LPC Vocoder model for low bit rate speech coding," in *IEEE Trans. Speech Audio Processing*, July 1995, vol. 3, pp. 242–250.

[10] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia application," in *Proc. ICASSP'00*, 2000.

[11] Y. Linde, A. Buzo, and R. M. Gray, "A algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, 1980.

[12] S. Srinivasan, D. Petkovic, and D. Ponceleon, "Toward robust features for classifying audio in the CueVideo system," in *Proc. 7th ACM Int. Conf. Multimedia*, 1999, pp. 393–400.

[13] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *J. VLSI Signal Process. Syst.*, June 1998.

[14] L. Lu, H. Jiang, and H. J. Zhang, "A robust audio classification and segmentation method," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 203–211.

[15] J. S. Boreczky and L. D. Wilcox, "A hidden Markov model frame work for video segmentation using audio and image features," in *Proc. ICASSP'98*, Seattle, WA, May 1998, pp. 3741–3744.

[16] J. N. L. Brummer, "Speaker recognition over HF radio after automatic speaker segmentation," in *Proc. IEEE South African Symp. Communications and Signal Processing (COMSIG-94)*, 1994, pp. 171–176.

[17] M. Sugiyama, J. Murakami, and H. Watanabe, "Speech segmentation and clustering based on speaker features," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1993.

[18] L. Wilcox, F. Chen, D. Kumber, and V. Balasubramanian, "Segmentation of speech using speaker identification," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1994.

[19] M. H. Siu, G. Yu, and H. Gish, "An unsupervised, sequential learning algorithm for the segmentation of speech waveform with multiple speakers," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1992, pp. 189–192.

[20] H. Gish, M. H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Proc. ICASSP'91*, 1991, pp. 873–876.

[21] A. Cohen and V. Lapidus, "Unsupervised speaker segmentation in telephone conversations," in *Proc. 19th Conv. Electrical Electronics Engineers in Israel*, 1996, pp. 102–105.

[22] K. Mori and S. Nakagawa, "Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition," in *Proc. ICASSP'01*, vol. I, 2001, pp. 413–416.

[23] S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[24] G. Schwarz, "Estimation the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.

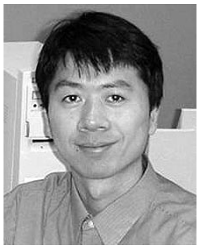
Lie Lu received his B.S. and M.S. from Shanghai Jiao Tong University, China, both in electrical engineering, in 1997 and 2000, respectively.

In 2000, he joined Microsoft Research Asia, Beijing, China, where he is currently an Associate Researcher with the Media Computing Group. His current interests are in the areas of pattern recognition, content-based audio analysis, and music analysis.

Hong-Jiang Zhang (S'90–M'91–SM'97) received the Ph.D. degree from the Technical University of Denmark and the B.S. degree from Zhengzhou University, China, both in electrical engineering, in 1982 and 1991, respectively.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at Massachusetts Institute of Technology Media Lab, Cambridge, MA, in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management; intelligent image processing and Internet media. In 1999, he joined Microsoft Research Asia, where he is currently a Senior Researcher and Assistant Managing Director in charge of media computing and information processing research. He has authored three books, over 200 referred papers and book chapters, seven special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as numerous patents or pending applications.

Dr. Zhang is a member of ACM. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences.



Hao Jiang received the Ph.D. in electronic engineering from Tsinghua University, Beijing, in 1999. He received the B.S. and M.S. degrees in electronic engineering from Harbin Engineering University, Harbin, China, in 1993 and 1995, respectively. He is currently pursuing the Ph.D. degree in computing science in Simon Fraser University, Vancouver, BC, Canada.

He had been an Associate Researcher with Microsoft Research Asia, Beijing, from 1999 to 2000. His current research interest is on multimedia, image and video processing, computer vision, and computer graphics.