Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 57-72

Robust Target Speaker Tracking in Broadcast TV Streams

Junmei Bai[*], Hongchen Jiang[*], Shilei Zhang[*],

Shuwu Zhang[*] and Bo Xu[*]

Abstract

This paper addresses the problem of audio change detection and speaker tracking in broadcast TV streams. A two-pass audio change detection algorithm, which includes detection of the potential change boundaries and refinement, is proposed. Speaker tracking is performed based on the results of speaker change detection. In speaker tracking, Wiener filtering, endpoint detection of pitch, and segmental cepstral feature normalization are applied to obtain a more reliable result. The algorithm has low complexity. Our experiments show that the algorithm achieves very satisfactory results.

Keywords: Speaker Tracking, Audio Segmentation, Entropy, GMM

1. Introduction

Broadcast TV programs are rich multimedia information resources. They contain large amounts of AV (audio & video) contents including speech, music, images, motion, text, and so on. Finding ways to extract and manage these various kinds of AV content information is becoming extremely important and necessary for application-oriented multimedia content mining and management. The analysis and classification of audio data are important tasks in many applications, such as speaker tracking, speech recognition, and content-based indexing. Among of them, target speaker tracking in TV streams is an important research topic for TV scene analysis. In contrast with general speaker recognition, speaker detection in audio streams usually requires segments of relatively homogenous speech and speaker tracking in this task should also determine the target speakers' locations, in other word, the starting and ending times. In such applications, effective methods for segmenting continuous audio streams

[*] The High-Tech. Innovation Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail: {jmbai, hcjiang, slzhang, swzhang, xubo}@hitic.ia.ac.cn

[Received June 15, 2005; Revised January 23, 2006; Accepted January 27, 2006]

into homogeneous segments are required.

The problem of acoustic segmentation and classification has become crucial for the application of automatic speech recognition to audio stream processing. The automatic segmentation of long audio streams and the clustering of audio segments according to different acoustic characteristics have received much attention recently [Lu and Zhang 2002; Chen and Gopalakrishnan 1998; Delacourt and Wellekens 2000; Wilcox *et al*. 1994; Pietquin *et al*. 2002]. To detect target speakers in an audio stream, it is best to segment the audio stream into homogeneous regions according to changes in speaker identity, environmental conditions and channel conditions. In fact, there are no explicit cues of changes among these audio signals, and the same speaker may appear multiple times in audio streams. Thus, it is not easy to segment an audio stream correctly. Various segmentation algorithms proposed in the literature [Lu and Zhang 2002; Chen and Gopalakrishnan 1998; Delacourt and Wellekens 2000; Ajmera *et al*. 2003; Cettolo and Federico 2000] can be categorized as follows [Chen *et al*. 1998]:

1) Decoder-guided segmentation algorithms: The input audio stream is first decoded by an automation speech recognition (ASR) systems, and then the desired segments are produced by cutting the input at the silence locations generated by the decoder. Other information from the decoder, such as gender information, can also be utilized in segmentation.

2) Model-based segmentation algorithms: Different models, e.g., Gaussian mixture models, are build for a fixed set of acoustic classes, such as telephone speech, pure music, etc, from a training corpus. In these schemes, a sliding window approach and multivariate Gaussian models are generally used. Decisions about the maximum likelihood boundary are made.

3) Metric-based segmentation algotithms: The audio stream is segmented at places where maxima of the distances between neighboring windows appear, and distance measures, such as the KL distance and the generalized likelihood ratio (GLR) distance [Fisher *et al*. 2003], are utilized.

These methods are not very successful at detecting acoustic changes that occur in data [Chen *et al*.1998]. Decoder-guided segmentation only places boundaries at silence locations, which in general have no direct connection with acoustic changes in the data. Model-based segmentation usually can not be generalized to unseen acoustic conditions. Meanwhile, both model-based and metric-based segmentation rely on a threshold which sometimes lacks stability and robustness. In addition, model-based segmentation does not generalize to unseen acoustic conditions.

As for target speaker detection, which is similar to general speaker verification, the traditional methods focus on likelihood ratio detection and template matching. Among these approaches, Gaussian Mixture Models (GMMs) have been the most successful so far

[Reynolds *et al.* 2000]. Reynolds also extended of these methods by adapting the speaker model from a universal background model (UBM). The speaker detector we adopted in our experiments is based on adapted GMMs. In the target speaker detecting system, we also used the segmental cepstral mean and variance normalization (SCMVN) to normalize the cepstral coefficients to get robust segmental parameter statistics that are suitable for various kinds of environmental conditions.

2. Overview

The task of automatic speaker tracking involves finding target speakers in test audio streams. Given an audio stream, all the segments containing a target speaker's voice must be located with the starting and ending times. The general approach to speaker tracking consists of three steps: audio segmentation, audio classification, and speaker verification. A complete block diagram of the proposed speaker tracking system is shown in Figure 1. The diagram shows how the components of the system fit together.

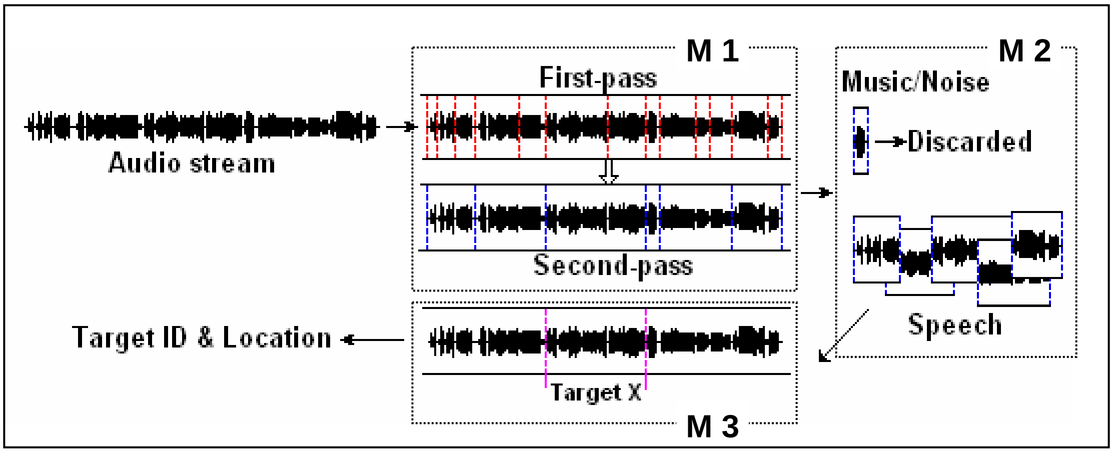


Figure 1. Block diagram of the speaker tracking system components

M1—Segmentation Module, **M2** —Classification Module, **M3**—Speaker verification

The three steps are defined as three modules in Figure 1, denoted as M1, M2, and M3. First, audio streams are segmented in M1 by means of two-pass audio segmentation. Then, in M2, these audio segments are classified into different classes, such as speech, music, noise and so on. Last, the speech segments are tested in M3 to verify if target speakers appear in the audio streams. Sometimes, M2 is not necessary when the speaker verification module can distinguish target speakers with other audio signals with acceptable precision. The individual blocks will be described in detail in following sections.

3. Two-Pass Audio Segmentation

The goal of automatic segmentation of audio signals is to detect changes in speaker identity, environmental conditions, and channel conditions. The problem is to find acoustic change detection points in an audio stream. A two-pass segmentation process for audio streams is presented in this paper. First, audio segmentation based on entropy is used to detect potential audio change points. Then, speaker change boundary refinement based on Bayesian decisions is applied.

3.1 First-Pass Segmentation Based on Entropy

In the first pass, we use entropy measures to determine the turn candidates. Entropy is a measure of the uncertainty or disorder in a given distribution [Papoulis 1991]. There are many methods for calculating entropy. Ajmera calculates entropy based on posterior probabilities and sets it as one of the features for discriminating speech and music [Ajmera *et al.* 2003]. It is a model-based classification scheme that makes decisions based on the scores of audio signals to two models: a speech model and a music model. Generally, the speech model is estimated from lots of speech spoken by different speakers, and it acts as a universal model. Thus, it is not suitable for distinguishing different speakers, particularly unknown speakers.

The entropy method used in this work is also an extension of the model-based segmentation scheme. Generally, model-based methods apply a maximum likelihood of the Gaussian process with a penalty weight to detect turns in audio streams. By appropriately defining this penalty, one can generate decisions based on the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), the Consistent AIC (CAIC), the Minimum Description Length (MDL) principle, and the Minimum Message Length (MML) principle. It has been found that BIC, MDL, and CAIC give the best results and that with proper tuning, all three produce comparable results [Cettolo *et al*. 2000].

In this paper, entropy is calculated based on statistical parameters of audio features. The decision rule is not based on scores but on the shape of the entropy contour. In order to clearly show the performance of our method, it is compared with BIC in this paper. The of entropy-based audio segmentation scheme is described in detail in the following:

Entropy of a Gaussian Random Variable [You *et al*. 2004]**:**

Assume a random variable X of dimension K. The entropy of the random variable (RV) is computed by first estimating its probability distribution function (pdf). We can compute the pdf either from the RV's histogram or from a parameterized distribution. The latter is used to reduce the amount of computation. Assume that the pdf follows a K-dimensional Gaussian density:

$$P(X) = |2\pi\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}, \qquad \text{(1-a)}$$

where μ is the mean vector and Σ is the covariance matrix. The entropy of X is

$$E(X) = -\int P(X) Log P(X) dX \quad . \qquad \text{(1-b)}$$

Eq. (1-b) can been replaced by [You *et al.* 2004]:

$$E(X) \approx KLog\sqrt{2\pi} + \log\Sigma \, . \qquad \text{(1-c)}$$

The entropy curve of a speech signal in a sliding window is calculated as follows:

Define $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_N\}$ as the cepstral sequence of an audio stream in a sliding window of N frames. At a given frame index j $(1 < j < N)$, the sliding window is partitioned into two sub-windows. Denote them as $\mathbf{Y}_{j(l)} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_j\}$ and $\mathbf{Y}_{j(r)} = \{\mathbf{y}_{j+1}, \mathbf{y}_{j+2}, \cdots, \mathbf{y}_N\}$, respectively. The lengths of the two windows are $N_{j(l)}$ and $N_{j(r)}$ ($N_{j(l)} + N_{j(r)} = N$) respectively. Assume that each window is generally modeled with a multivariate Gaussian density, such as $\mathbf{N}(\boldsymbol{\mu}_{j(l)}, \boldsymbol{\Sigma}_{j(l)})$ and $\mathbf{N}(\boldsymbol{\mu}_{j(r)}, \boldsymbol{\Sigma}_{j(r)})$, respectively. The sum of the entropy of each side of the window is computed as follows:

$$E_{j(l)} = \sum_{i=1}^{N_{j(l)}} (K\log\sqrt{2\pi} + \Sigma_{j(l)}{}^{(i)}) = \sum_{i=1}^{N_{j(l)}} K\log\sqrt{2\pi} + N_{j(l)}\Sigma_{j(l)}) \, , \qquad \text{(1-d)}$$

$$E_{j(r)} = \sum_{i=1}^{N_{j(r)}} (K\log\sqrt{2\pi} + \Sigma_{j(r)}{}^{(i)}) = \sum_{i=1}^{N_{j(r)}} K\log\sqrt{2\pi} + N_{j(r)}\Sigma_{j(r)} \, . \qquad \text{(1-e)}$$

Then, the segmentation entropy at j can be computed as follows

$$\begin{aligned} E(j) &= \sum_{i=1}^{N_{j(l)}} K\log\sqrt{2\pi} + N_{j(l)} \times \log|\boldsymbol{\Sigma}_{j(l)}| + \sum_{i=1}^{N_{j(r)}} K\log\sqrt{2\pi} + N_{j(r)} \times \log|\boldsymbol{\Sigma}_{j(r)}|, \\ E(j) &= NK\log\sqrt{2\pi} + N_{j(l)} \times \log|\boldsymbol{\Sigma}_{j(l)}| + N_{j(r)} \times \log|\boldsymbol{\Sigma}_{j(r)}| \, . \end{aligned} \qquad \text{(1-f)}$$

$NK\log\sqrt{2\pi}$ is a constant. It is ineffective for determining the entropy curve and can been omitted. Thus, the segmentation entropy at j can be simplified as follows:

$$E(j) = N_{j(l)} \times \log|\boldsymbol{\Sigma}_{j(l)}| + N_{j(r)} \times \log|\boldsymbol{\Sigma}_{j(r)}| \, . \qquad \text{(1)}$$

Decision making is performed by analyzing the entropy curve in each window as described below.

H1: There is a potential change point in the sliding window.

The sequence entropy value shows a step-down change until it reaches a minimal value at time t . Then, it increases gradually. t can be considered as a change point. Here, $t = \arg\min_j E(j)$.

H0: There is no any change point in the sliding window.

The segmentation entropies vary randomly.

We can make the following observations:

a) The minimal entropy varies for different window sizes and different audio conditions. However, if the entropy decreases gradually till it reaches a minimal polar, then it increases gradually, there is a changing point at the polar.

b) Since there are fewer data in the region close to the original point on the left, the segmentation entropies in this region are unable to describe the entropy curve accurately. The same is true, on the right. Thus, these two regions are ignored in the final analysis. As shown in Figure 2, θ_t is defined as the number of the points ignored on each side.

c) The basic processing unit or the sliding window length is 3s; however, the overlapping length between two neighboring windows is not fixed. If there is not change point in the prior window, the overlapping length is 1.5s; otherwise, the overlapping length is relative to the location of the last change point in the prior window.

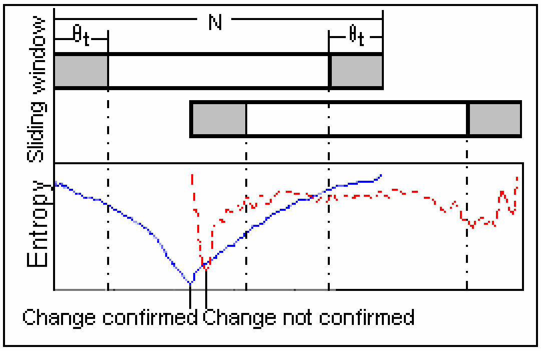


Figure 2. Samples of entropy contour

3.2 Second-Pass Speaker Change Boundary Refinement

Often there are false positives in potential speaker change points obtained with the algorithms described above. To remove false positives, a refinement algorithm is applied. The algorithm is based on the dissimilarity between two adjacent sub-segments. In this step, two distance measures, the Bayesian decision and KL distance, are applied to validate or discard candidates from the first pass. Suppose the feature vector extracted from each sub-segment is Gaussian, and assume that the feature probability distribution functions are n-variable normal populations, such as $\mathbf{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathbf{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. The Bayesian decision distance between two speech segments can be defined as [Lu *et al*. 2002]

$$D_{BD} = \frac{1}{2} tr[(\Sigma_1 - \Sigma_2)(\Sigma_1^{-1} - \Sigma_2^{-1})] . \tag{2}$$

Provided that the speech of each segment can been modeled with a multivariate Gaussian density, the Kullback-Leibler (KL) distance between two speech slices is defined by [Homayoon *et al*. 1998]

$$D_{KL} = \frac{\sum_{i=1}^{M}(w_1^i d_1^i + w_2^i d_2^i)}{\sum_{i=1}^{M}(w_1^i + w_2^i)}, \tag{3}$$

$$d_{ij} = \frac{\Sigma_1^i}{\Sigma_2^j} + \frac{\Sigma_2^j}{\Sigma_1^i} + \frac{\mu_1^i - \mu_2^j}{\Sigma_1^i} + \frac{\mu_1^i - \mu_2^j}{\Sigma_2^j}, \tag{3-a}$$

$$d_1^i = \min_j (d_{ij}), \tag{3-b}$$

$$d_2^j = \min_i (d_{ij}). \tag{3-c}$$

$w_t\{w_t^{\,i} \mid i = 1, 2, ..., M\}$ is the mixture weight of the model of the *t*th segment.

In general, if two speech segments are spoken by the same speaker, the distance between them will be small; otherwise, the distance will be large. Thus, we apply a simple criterion: if the distance between two speech segments is larger than a given threshold, then these two segments can be considered as to be spoken by different speakers. The thresholds adopted in this study were set experientially. Figure 3 shows an example of two-pass audio segmentation of 26-second long audio stream. The audio stream includes two speakers and 3 speaker change boundaries, which are 7s, 15s and 22s respectively. It can be seen that the number of the potential boundaries is greater than that of real boundaries. The Bayesian decision is performed on these potential speaker change points to remove the false ones. In Figure 3, D_{bd},

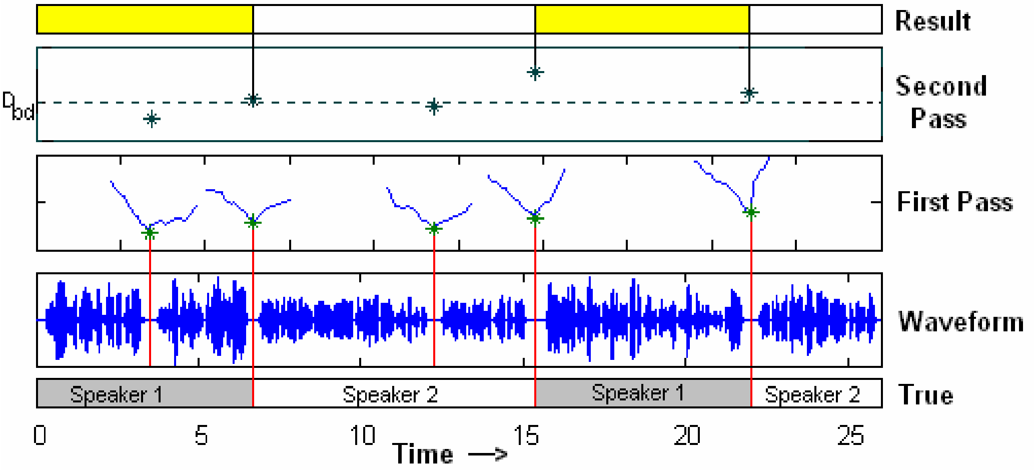


Figure 3. Two pass segmentation procedure: the entropy contour and the Bayesian decision

is an experiential threshold for the Bayesian decision.

3.3 Audio Segments Classification

The aim of audio classification is to distinguish speech and other audio signals. Currently, the state-of-the-art method of classification is based on GMM. Four models were applied in our experiments, a speech model, an unvoiced model, a music model, and a noise model, to classify the audio segments. Among them, only speech slices were used to detect target speakers in subsequent processing.

4. Target Speaker Tracking System

To a certain extent, speaker detection is similar to automatic speaker verification (ASV), which is used to verify the identity claimed by a speaker. The general approach to speaker detection mainly consists of four parts: speech signal pre-processing, speaker feature extraction, speaker modeling, and recognition.

4.1 Speech Slice Pre-Processing

In automatic speaker detection systems, the mismatch between training and recognition, generated by additive or convoluting noises, often severely degrades the recognition accuracy. In addition, the non-speech signals, mainly silence and noise, contain little information of speakers. They are the same for each person and contain no distinguishing features, only ones that are confusing for speaker detection. They can degrade the discrimination ability for different speakers. Thus it is necessary to reduce the noise and discard the irrelevant information before performing speaker features extraction. In our experiments, we applied Wiener filtering and pitch-based endpoint detection in speech slice pre-processing.

Though pitch is a robust feature to noise, it is difficult to measure pitch accurately and reliably for several reasons. Since the key is to detect the active endpoints by means of pitch,

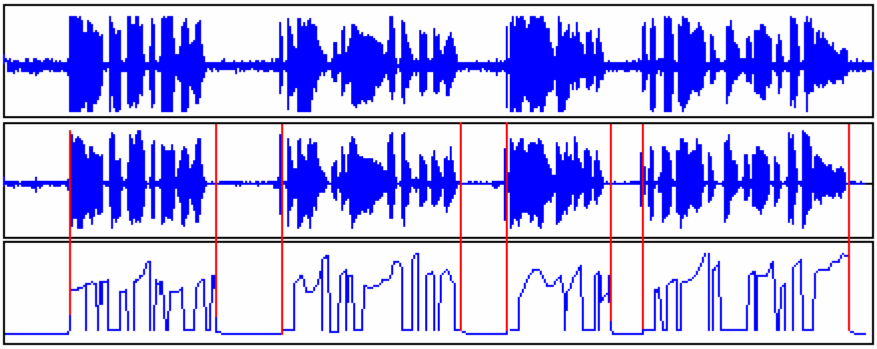


Figure 4. Pre-processing by Wiener filter and endpoint detection on pitch

it is not appropriate to put much emphasis on the precision values of pitch. Moreover, we use wiener filter to alleviate the noise, which makes the pitch detection more precise. In Figure 4, we can see that the pitch, which is mostly susceptible to noise, is near the endpoint. We set the active endpoint at the place where the pitch is less than zero. Although the pitch may not be precise, it is valid for endpoint detection. If the interval between two adjacent unvoiced frames is too short, say, less than 10 frames, then these unvoiced frames will be reserved.

4.2 Speaker Feature Extraction and Normalization

Although there is no exclusive feature for distinguishing different speakers' voices, the speech spectrum has been shown to be very effective for speaker recognition. This is because the spectrum reflects a person's vocal tract structure, the predominant physiological factor that distinguishes one person's voice from others. The Mel-frequency cepstral coefficient (MFCC) vectors have been used extensively for speaker recognition. However, the MFCC features can be severely affected by noise. Thus, some methods should be used to compensate for the corrupted speech.

The widely used method for Cepstral feature normalization is Cepstral mean subtraction (CMS). CMS is performed over an entire file, and it can reduce the stationary convolution noise caused by the channel. However, CMS can also reduce some slow dynamic features of speakers. In this study, the segmental cepstral mean and variance normalization (SCMVN) were used. SCMVN is calculated as follows:

$$\hat{x}_{t+(L-1)/2}(i) = \frac{x_{t+(L-1)/2}(i) - \mu_t(i)}{\sigma_t(i)} \quad , \tag{4}$$

where, X_t is the feature vector at time t, and L is the length of the sliding window; t, which is the first frame in the current window, gives the current place of the window in the speech; $\mu_t(i)$ and $\sigma_t(i)$ are the means and variances of the feature vector in this window. It should be noted that the length of the window, L, is fixed since the normalization of all feature should be uniform. In addition, a proper value of L should be adopted. The estimations of $\mu_t(i)$ and $\sigma_t(i)$ may be imprecise if L is too short. And if it is too long, the calculation will be more complex.

SCMVN has two possible effects: Firstly, it can reduce the action of addition noises in feature variance. Generally, addition noises result in decreased variance. Secondly, the features are mapped to a normal distribution over a sliding window, which is helpful for modeling the speakers' GMM later in speaker recognition.

4.3 Speaker Tracking

The basic speaker detector is a likelihood ratio detector with target and alternative probability distributions. For text independent speaker verification GMMs (Gaussian Mixture Models) have been most successful so far [Reynolds *et al*. 2000]. The test ratio may be expanded by using the Bayesian rule:

$$T(x) = \frac{f(\lambda_i \mid x)}{f(\lambda_{UBM} \mid x)} = \frac{g(\lambda_{UBM})\, f(x \mid \lambda_i)}{g(\lambda_i)\, f(x \mid \lambda_{UBM})}, \tag{5}$$

where $g(\lambda)$ is the prior density. In fact, the prior density is assumed to be equal for the UBM and the target model. The set of feature vectors is often very large, hence, the value of $f(..)$ is often very small. Therefore, it is common to compute the logarithm of the test ratio instead. The log-test ratio is given by

$$\theta_i(x) = \log f(x \mid \lambda_i) - \log f(x \mid \lambda_{UBM}) \,. \tag{6}$$

Thus, the most suitable speaker models can be found based on the largest likelihood ratio. If the largest likelihood ratio is larger than a threshold, the identity of the current speaker can be determined; otherwise, the current segment is considered for a new speaker. In this way, we can determine the identity of the current speaker. Suppose that so far, K speakers are registered in the speaker model database; the concrete expression for identifying the speaker of the current segment is as follows:

$$ID = \begin{cases} \arg\max\limits_i \theta_i & if \; \max \theta_i \ge \theta_0 \\ \quad Non & if \; \max \theta_i \le \theta_0 \end{cases}, \tag{7}$$

where $1 \le i \le K$ and *Non* represents a new speaker. The threshold θ_0 can be either speaker dependent or speaker independent. The purpose of speaker dependent thresholds is to reduce the negative effects of speaker dependent variability on performance. Another solution is to apply a reversible transform to score values so that the result is equivalent to using speaker dependent thresholds. For practical reasons, the transform is based on impostor scores rather than on true speaker scores. One such method, currently known as znorm [Reynolds 1995], transforms the impostor score distribution to zero mean and unit variance, while a Gaussian distribution is assumed. For an observation x and a claimed identity λ_i, the normalized log-test is given by

$$\theta_i^{Znorm}(x) = \frac{\theta_i(x) - \mu_i}{\sigma_i}, \tag{8}$$

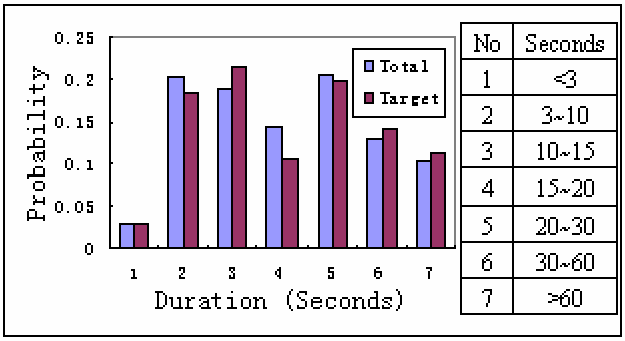
where μ_i and σ_i are the moment estimates of the impostor score distribution for a speaker λ_i.

5. Experiments

5.1 Database

The proposed audio segmentation and speaker tracking algorithms were evaluated using an audio database, recorded directly from the CCTV news channel. The database is composed of about 10 hours of audio streams, which are from different TV programs, such as news, interviews, music, and movies. In the test database, at least one target speaker appeared in each file.

Figure 5 reports the length statistics for the segments in the test set. A segment was defined as a contiguous portion of an audio signal, homogeneous in terms of acoustic source and channel. The duration of two adjacent turns in the test data varied from 2 seconds to 5 minutes. In Figure 5, the x-axis is the time duration, and the number reprensents the duration. On the right side of Figure 5, the first row corresponds to the second row. For example, 1=”<3s” and 2=”3s~10s”. This shows that about 2% of the audio segments were less than 3 seconds long. We tested the performance with windows of 2 seconds and 3 seconds. It was observed that the performance decreased dramatically when the two-second window is used. Thus, we selected 3 seconds as the unit window size. That is to say, for those speaker segments which were less than 3 seconds long, the segmentation results were not reliable.



No	Seconds
1	<3
2	3~10
3	10~15
4	15~20
5	20~30
6	30~60
7	>60

Figure 5. Histogram for the audio segment durations of all audio streams and Target speakers

5.2 Experimental Setup

The input audio stream was first down-sampled into a uniform format: 8KHZ, 16bits, and mono-channel, regardless of the input format. In first pass segmentation, the speech stream was then pre-emphasized and divided into sub-segments using 3-second window with some overlapping. That is, the basic processing unit was 3 seconds; however, the temporal resolution of segmentation was not fixed. If there was no change point in the prior window, the overlapping length was 1.5 second, or the overlapping length was relative to the location of the last change point in the prior window.

In target speaker detection, the most important features extracted from the frame were MFCC and pitch. MFCC and the delta parameters were employed to characterize target speakers. The 16-dimensional MFCC vector and 1-dimensional energy were extracted from the speech signal every 12 ms with a 24 ms window. The delta parameters were then computed and appended to the previous vectors, thus producing a 34-dimensional feature vector.

There were a total of 40 target speakers, who consisted of reporters, commentators, comperes, and interviewees. The target models were adapted from UBM parameters, using two minutes of training data. The target speaker detector was a likelihood ratio detector for adaptation GMMs. Our UBM was a 1024 mixture GMM, trained using about 6 hours of broadcast data from 60 speakers with equal number of males and females. Target models were derived by means of Bayesian adaptation from the UBM parameters using two minutes of training data. Only the mean vectors were adapted, as this had been observed to provide better performance. The amount of adaptation of each mixture mean was data dependent.

The baseline system only used CMS to alleviate noises; then, Wiener filtering, endpoint detection via the pitch, and SCMVN were applied, respectively.

5.3 Experimental Results

The criteria of performance for audio segmentation and speaker detection are shown below:

For audio segmentation, the false alarm rate (FAR) and missed detection rate (MDR) were calculated as follows [Lu *et al*. 2002]:

$$FAR = \frac{number \quad of \quad false \quad \det ection}{number \quad of \quad false \quad \det ection + number \quad of \quad true \quad change} \times 100\% \quad ,$$

$$MDR = \frac{number \quad of \quad miss \quad \det ection}{number \quad of \quad true \quad change} \times 100\% \quad .$$

For target speaker detection: the false alarm rate (FA), false reject rate (FR), and Equal Error Rate (ERR) were calculated as follows:

$$FA = \frac{number \quad of \quad false \quad accepted \quad as \quad t\arg ets}{number \quad of \quad segments \quad - \quad number \quad of \quad t\arg et \quad segments} \times 100\% \quad ,$$

$$FR = \frac{number \quad of \quad False \quad rejected}{number \quad of \quad true \quad t\arg et \quad segments} \times 100\% \quad .$$

When $FA = FR$, $ERR = FA = FR$. ERR is a common criterion for judging the performance of speaker verification systems.

5.3.1 Results of Audio Segmentation

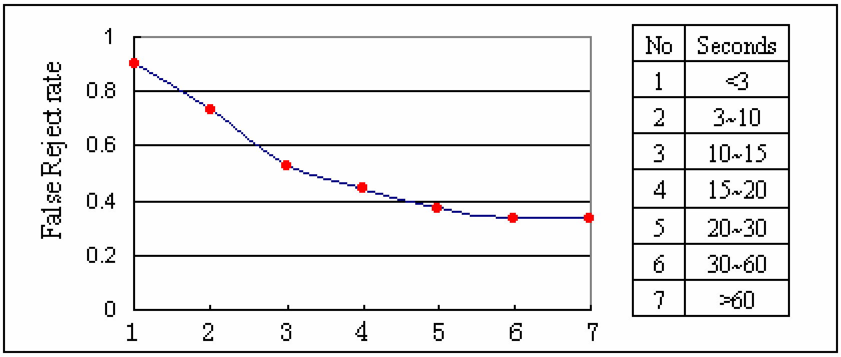
The statistics results of audio segmentation are shown in Table 1. In first pass segmentation, the entropy-based method was better than BIC, particularly in *MDR*. However, *FAR* was still a little high with both methods. This was mostly due to the following reasons. First, *FAR* in long segments is great. As shown in Figure 5, about 10% of the segments were longer than 60 seconds. These long segments resulted in 5%-10% *FAR*. Second, the noise information increased *FAR*. In fact, some of the false detections in long segments affected the speaker-tracking performance a little, for about 20 seconds of speech is enough for speaker recognition. What's more, about 25% *FAR* appeared in speech signals. Thus, a speaker change boundary refinement algorithm was applied to remove false positives. As shown in Table 1, second pass refinement decreased *FAR* from 30.4% to 14.4% and from 31.2% to 14.9% based on the entropy results and on BIC results, respectively, In *MDR*, there was about a 0.6% increase based on the entropy results and a 1.8% increase based on the BIC results. As for the second pass refinement schemes, Bayesian decision was little better than the KL distance.

Table 1. The results of two-pass audio segmentation

		FAR	**MDR**			**FAR**	**MDR**
First Pass	Entropy	30.4%	6.5%	**Second pass**	BD	**14.4%**	**7.1%**
					KL	16.0%	7.3%
	BIC	31.2%	13.1%	**Second pass**	BD	14.9%	14.5%
					KL	15.2%	15.0%

5.3.2 Results of Target Speaker Detection

There are many factors that affect the performance of speaker detection. Among them, the target speech duration is a very important factors especially for the false reject (*FR*) rate in target speaker detection. Generally speaking, the shorter the speech is, the higher *FR* and *FA* will be. As shown in Figure 6, the FR rate decreased greatly with increasing time when the speech durations were less than 20 seconds long. And it changed little when the speech durations were longer than 20 seconds. Noise is another interference factor in target speaker detection. The performance in target speaker detection with different strategies is shown in Table 2. The EER and the relative improvement compared with the baseline are illustrated in Table 2. Compared with the conventional CMS, SCMVN was better at compensating for the corruption caused by noise. Its effect was clear in target speaker detection. Wiener filtering and endpoint detection based on pitch are only used in speaker detection because the error in noise estimating in Wiener filtering increases when the noise environment changes, so it cannot work well with long speech durations. In this case, Wiener filtering is not helpful but costly in terms of time. And silence signals are useful for audio segmenting, so they are not discarded. However, their effects in speaker detection were clear in our experiments. The integrated system with SCMVN, Wiener filtering, and endpoint detection showed the best performance.



No	Seconds
1	<3
2	3~10
3	10~15
4	15~20
5	20~30
6	30~60
7	>60

Figure 6. The FR of speaker detection at different speech durations

Table 2. The ERR of target speaker detection

Case	ERR	ERR Relative Reduction
Baseline	25.2%	0
WF + ED	23.3%	9.1%
SCMVN + WF + ED	22.8%	9.5%

6. Conclusion

In this paper, we have presented a novel approach to unsupervised audio segmentation and a speaker tracking system. A two-pass audio change detection algorithm has been proposed, which includes potential audio change detection and speaker boundary refinement. The results of two-pass audio segmentation are classified as speech or music according their characteristics. Speaker tracking is based on the results of audio classification. In speaker tracking, Wiener filtering, endpoint detction based on pitch, and the segmental cepstral mean and variance normalization are applied to get more reliable results. The algorithm achieves satisfactory accuracy.

There is still room for improvement of the proposed approach. In the experiments, we found that if two speakers were speaking synchronously, it was not easy to detect the change boundary. It was also found that the same speaker in various environments sometimes was detected as different speakers or rejected. This indicates that our compensation for the mismatch effect of the environment or channel is still insufficient. In our future research, we will focus on these issues.

Acknowledgement

Support provided by the National Natural Science Foundation of China (NSFC) under grant no. 60475014 and the National Hi-tech Research Plan under grant no. 2005AA114130 is gratefully acknowledged.

Reference

Ajmera, J., I. McCowan, and H. Bourlard, “Speech/music segmentation using entropy and dynamism features in a HMM classification framework,” *Speech Communication*, 40(3), 2003, pp.351-363.

Bai, J., S. Zhang, R. Zheng, S. Zhang, and B. Xu, “Audio Segmentation and Speaker Detection in Broadcast TV Stream,” In *Proc. of 10th International Conference on SPEECH and COMPUTER*, 2005, Patras, Greece, pp.547-550.

Beigi, H. S. M., S. H. Maes, and J. S. Sorensen, “A Distance Measure Between collections of Distributions and Its Application to Speaker Recognition,” In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing, 1998*, Seattle, Washington, USA, pp. 753-756.

Campbell, J.P., “Speaker Recognition: a Tutorial.” *Proceedings of The IEEE*, 85(9), 1997, pp. 1437-1462.

Cettolo, M., and M. Federico., “Model Selection Criteria for Acoustic Segmentation,” In *Proc. of the ISCA ITRW ASR2000 Automatic Speech Recognition*, 2000, Paris, France, pp. 221-227.

Chen, S., and P.S. Gopalakrishnan, “Speaker, Environment, and Channel Change Detection and Clustering via the Bayesian Information Criterion,” In *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*. 1998. Virginia, USA, pp.127-132.

Delacourt, P., and C.J. Wellekens, “DISTBIC: A speaker-based segmentation for audio data indexing,” *Speech Communication*, 32 (1-2), 2000, pp.111-126.

Dunn, R.B., D. A. Reynolds, and T. F. Quatieri, “Approaches to speaker detection and tracking in conversational speech,” *Digital Signal Processing*, 10 (1-3), 2000, pp.93–112.

Fisher, E., J. Tabrikian, and S. Dubnov, “Generalized likelihood ratio test for voiced/unvoiced decision using the harmonic plus noise model,” In *Proc. Of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2003, Hong Kong, pp. 440-443.

Jin, H., F. Kubala, and R. Scwartz, “Automatic Speaker Clustering,” In *Proc. of the DARPA Speech Recognition Workshop*, 1997, pp. 108-111.

Lu, L., and H.J. Zhang. "Speaker change detection and tracking in real-time news broadcasting analysis, " In *Proc. of the 10th ACM International Conference on Multimedia*, 2002, Juan-les-Pins, France, pp. 602-610.

Mori, K., and S. Nakagawa. "Speaker Change Detection and Speaker Clustering Using VQ Distortion for Broadcast News," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2001, Salt-Lake City, USA, pp.413-416.

Papoulis, *A Probability, Random Variables, and Stochastic Processes*. 3rd ed. McGraw-Hill, 1991.

Pietquin, O., L. Couvreur, and P. Couvreur, "Applied Clustering for Automatic Speaker-based segmentation of Audio Material," *Belgian Journal of Operations Research, Statistics and Computer Science*, 41(1-2), 2002, pp. 69-81.

Reynolds, D. A., T.F. Quatieri, and R.B. Dunn, "Speaker Verification. Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, 10(1-3), 2000, pp. 19-41.

Reynolds, D.A., "Speaker Identification and Verification using Gaussian Mixture Speaker Models," *Speech Communication*, 17(1-2), 1995, pp. 91-108.

Sigler, M.A., U. Jain, B. Raj, and M. Stern. "Automatic Segmentation Classification and Clustering of Broadcast News Audio," In *Proc. of the DARPA Speech Recognition Workshop*, 1997, pp. 97-99.

Tsekeridou, S., and Ioannis Pitas, "Audio-Visual Content Analysis for Content-Based Video Indexing," In *Proc. of 1999 IEEE Int. Conf. on Multimedia Computing and Systems*, 1999, Florence, Italy, pp. 667--672.

Wilcox, L., F. Chen, D. Kumber, and V. Balasubramanian, "Segmentation of Speech Using Speaker Identification," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 1994, Adelaide, Australia, pp.161-164.

Wu, J., J. Droppo, L. Deng, and A. Acero, "A noise-robust ASR Front-end Using Wiener filter Constructed from MMSE Estimation of Clean Speech and Noise," In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, St. Thomas, U.S, 2003, pp.321-326.

You, H., Q. Zhu, and A. Alwan, "Entropy-based variable frame rate analysis of speech signals and its application to ASR," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2004, Montreal, Canada, pp. 529-552.