Unsupervised Speaker Indexing Using Generic Models

Soonil Kwon, *Student Member, IEEE,* and Shrikanth Narayanan, *Senior Member, IEEE*

***Abstract*—Unsupervised speaker indexing sequentially detects points where a speaker identity changes in a multispeaker audio stream, and categorizes each speaker segment, without any prior knowledge about the speakers. This paper addresses two challenges: The first relates to sequential speaker change detection. The second relates to speaker modeling in light of the fact that the number/identity of the speakers is unknown. To address this issue, a predetermined generic speaker-independent model set, called the sample speaker models (SSM), is proposed. This set can be useful for more accurate speaker modeling and clustering without requiring training models on target speaker data. Once a speaker-independent model is selected from the generic sample models, it is progressively adapted into a specific speaker-dependent model. Experiments were performed with data from the Speaker Recognition Benchmark NIST Speech corpus (1999) and the HUB-4 Broadcast News Evaluation English Test material (1999). Results showed that our new technique, sampled using the Markov Chain Monte Carlo method, gave 92.5% indexing accuracy on two speaker telephone conversations, 89.6% on four-speaker conversations with the telephone speech quality, and 87.2% on broadcast news. The SSMs outperformed the universal background model by up to 29.4% and the universal gender models by up to 22.5% in indexing accuracy in the experiments of this paper.**

***Index Terms*—Generic models, localized search algorithm (LSA), Markov chain Monte Carlo (MCMC) method, maximum a posteriori (MAP), sample speaker models (SSM), universal background model (UBM), universal gender models (UGM), unsupervised speaker indexing.**

I. Introduction

RECOGNIZING speakers from the speech signal is one of the key challenges in biometrics. Speaker recognition encompasses speaker verification and speaker identification. Speaker verification is to verify a person's claimed identity from his (or her) voice. In speaker identification, without a prior identity claim, the system decides who the person is. There are two categories: text-dependent and text-independent speaker recognition. In text-dependent recognition, the spoken phrase is known to the system whereas in the text-independent case, the spoken phrase is unknown. Speaker indexing in general is a text-independent speaker identification problem [1], [2].

Speaker indexing, the process of determining who is talking when, is an integral element of speech data monitoring and content-based data mining applications. Consider, for example, applications such as meeting/teleconference monitoring, archiving and browsing. A key motivation arises from the fact that it is impossible or tedious to attend all relevant meetings face to face. Multimedia meeting or teleconference monitors and browsers can be useful for conveniently obtaining meeting information, such as who is saying what and when, remotely through on-line or off-line systems [3], [4]. Specially, these applications commonly include a speaker indexing process that tags speaker-specific portions of data to pin point who is talking when [5].

Speaker indexing can be divided into two categories based on processing requirements: on-line and off-line. Both on-line and off-line indexing can be executed only sequentially, a characteristic of speaker indexing. Off-line speaker indexing can be used for record keeping, but it is not suitable for real-time meeting or teleconferencing systems that demand on-line processing. One of the main technical differences between off-line and on-line speaker indexing is the feasibility of multipass processing over the same data. Hence, in off-line indexing, it is possible to use various speaker indexing algorithms at each iteration. However, in on-line indexing, only one strategy can be used through the whole sequential process. Recently we proposed an on-line method that picks out the speech segments from an audio stream and classifies them by speakers [6].

To enable speaker indexing, ideally we need information about speakers such as the number of speakers and the appropriate speaker models. However, in some scenarios, it is not easy to obtain *a priori* information about the target speakers in the data, including the number of speakers, in advance. Consider for example speaker indexing applied to broadcast news (interviews). It may not be easy to obtain information about the reporters and interviewees in advance. Hence, unsupervised speaker indexing may be required. Assuming one is using streaming audio, we are limited to making any indexing decision with only current and previously seen speech data from the session. Furthermore, since the models of speakers are not available *a priori* for indexing, we need to create and update them on the fly. This leads to a number of challenges. In general, under these circumstances of sequential learning, data are not sufficient to build a speaker model initially. Although a model can be roughly built, it is apt to cause decision errors due to potential uncertainty in the unsupervised learning. To address the problem, we need some method to enable effective model bootstrapping [7].

Manuscript received January 20, 2004; revised August 2, 2004. This work was supported in part by funds from the National Science Foundation (NSF) through the Integrated Media Systems Center, an NSF Engineering Research Center, under Cooperative Agreement EEC-9529152 and a CAREER award, the Department of the Army under Contract DAAD 19-99-D-0046, and the ARDA VACE project. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Geoffrey Zweig.

The authors are with the Department of Electrical Engineering and IMSC, University of Southern California, Los Angeles, CA 90089 USA (e-mail: soonilkwan@alumni.usc.edu; shri@sipi.usc.edu).

Digital Object Identifier 10.1109/TSA.2005.851981

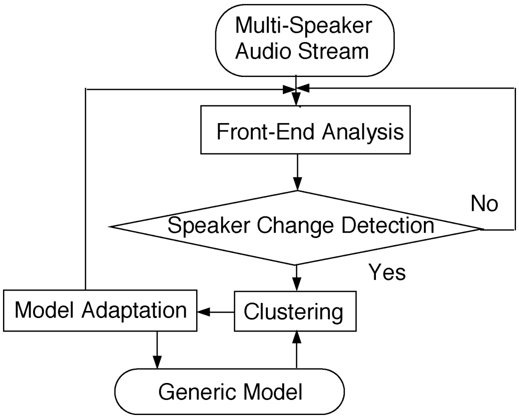

Fig. 1. Block diagram of the unsupervised speaker indexing process with generic models.

There are two kinds of generic models that have been proposed previously for this purpose: universal background models (UBM) and universal gender models (UGM). In this paper, we propose a new method for creating and evaluating generic models, referred to as the sample speaker models (SSM). This is built on the hypothesis that a speech data corpus, independent from the target data, can help initialize a model set for unsupervised speaker indexing. Samples are picked from a pool of generic speaker models using the Markov chain Monte Carlo (MCMC) method. The sample model set is predetermined at training. It is important to note that the speakers in the training data are independent of the testing data. In other words, the generic model set can be used for initializing/bootstrapping any speaker indexing process, and can be referred to during speaker clustering with the target test data. After clustering, a selected model can be continually adapted with the test data that are used for clustering [Fig. 1]. The model adaptation step in this paper uses the maximum *a posteriori* (MAP) scheme.

Before clustering, detection of speaker changing points is needed. This step sequentially binds data segments according to speakers that helps to improve the performance of indexing. There are two issues associated with speaker change detection implementation: the size of analysis segments and the specific analysis approach. The size of analysis segments is usually fixed. A large data analysis segment size is useful toward an improved correct indexing decision, as it includes more information about the speakers for indexing. However, it is apt to miss any speaker changes that may occur within an analysis segment. To solve this problem, a smaller data analysis segment size can be used, but it requires a robust speaker change detection process to improve the precision [6]. To detect the changing points robustly, sufficient overlapping across the analysis frames is required, resulting in higher computational complexity. Without sufficient overlapping, changing points are easily missed. The localized search algorithm (LSA) is adopted as a compromise between these conflicting requirements. Section III will provide details about this algorithm. We use the generalized likelihood ratio (GLR) Test for speaker change detection. Though the GLR test can be unstable for small amounts of analysis data, clustering can help compensate for this instability.

Several efforts have been reported on speaker indexing. Methods based on speaker verification using speaker subspace for speaker indexing were proposed by Nishida and Ariki [9]. In this paper, a speaker model was initialized and then the next speech segment was verified if it was from the same speaker as the first one. They used only 1-s segments, and these were too short to build an initial speaker model. Some segments including the speech of more than two speakers could not be correctly clustered without the speaker change detection. Rosenberg *et al.* used the GLR test for initial segmentation of speaker indexing. After initial segmentation, speaker models were constructed and then repeatedly segmented. Their process focused on the iterative segmentation and clustering that was only for the off-line speaker indexing systems [5]. Solomonoff introduced the metric based on purity and completeness of clusters for speaker clustering. With this method, even though it is not necessary to train speaker models, it is not found to be robust to environmental noises [10]. There are other efforts that have been reported on on-line speaker segmentation and clustering without prior knowledge of speakers and speaker models. The UBM was used to classify feature vectors by Wu [11]. Liu used the hybrid speaker clustering method, which utilized both the dispersion and GLR threshold [12].

In many previous scenarios of on-line speaker indexing or clustering, there is no prior knowledge about the identity or the number of speakers involved. The speaker indexing and model construction can be performed sequentially without storing all the testing data in advance. However, the problem is that sequentially constructed models may not represent speakers well due to model initialization problems. In addition, when the training of speakers is not supervised, this problem also potentially leads to continual error propagation. Without good initial models for speaker indexing, we cannot effectively build/update speaker models sequentially and incrementally. Recall that sequentially constructed models in the unsupervised indexing scenario cannot represent speakers well due to the small initial amount of data. We try to solve this critical drawback by employing an alternative method of using the notion of generic speaker models.

The experiments in this paper were conducted on data from the "Speaker Recognition Benchmark NIST Speech Corpus" (1999) and the "HUB-4 Broadcast News Evaluation English Test Material" (1999). We performed two main experiments with these data: model adaptation/convergence and speaker indexing. The speaker indexing evaluation, that explores the performance of the generic models, included three tests with different speech materials: two-speaker conversations, four-speaker conversations, and broadcast news. The experimental results showed that our on-line (sequential, causal) unsupervised speaker indexing can achieve a recognition rate comparable with a state of the art off-line system [5]. It also showed higher accuracy compared to other generic models such as the UBM and UGM under the various experimental conditions: two-speaker conversations, four-speaker conversations, and broadcast news.

The rest of this paper is organized as follows: Section II explains our unsupervised speaker indexing system; Section III describes speaker change detection using the LSA; Section IV introduces the notion of generic models, and explores bootstrapping with these generic models; Section V discusses clustering and model adaptation; Section VI and Section VII describe our experiments and results, respectively. Conclusions and future plans are described in Section VIII.

II. Unsupervised Speaker Indexing

A block diagram of the speaker indexing process is shown in Fig. 1. The first step is front-end analysis where the incoming audio samples are classified into foreground speech and other background audio (noise) types. Generally audio data can be categorized into four broad classes: speech, music, environmental noise, and silence. In speaker indexing, we only need speech/nonspeech discrimination. When there is background noise or music, it is likely to be overlapped with speech. Corrupted speech is not easily discriminated from noise. Since it is critical that we should not lose any speech data, the focus of the classification is to minimize false rejection, perhaps, even at the cost of false acceptance. Usually, for speech/nonspeech discrimination, a zero-crossing rate and short-time energy are used. It is well known that speech has a higher level of variation in the zero crossing rate [8].

Only the speech data are used for the next step, speaker change detection. In this step, the system sequentially detects whether a speaker change occurs in the middle of a speech analysis frame, without assuming any specific knowledge about speakers. Some of the challenges faced by this detection problem are addressed in Section III. Once the speaker change detection determines the boundary, all the data between the speaker change points are used for speaker clustering. In the clustering step, we use speaker models from a predetermined generic model set. After clustering, the speaker-independent generic model is adapted into an appropriate speaker dependent model during the indexing process. The adapted model is replaced with the original model before adaptation or inserted back in the generic model set. When new audio samples after the boundary of the current speaker come into the system, the previous steps are repeated until all data are exhausted.

III. Speaker Change Detection Using Localized Search Algorithm (LSA)

A. *Speaker Change Detection*

Robust speaker change detection is a critical prerequisite for speaker clustering. If we falsely detect a speaker changing point, we may compensate for the error through the speaker clustering step. However, if we totally skip the real changing point, the clustering step cannot recover it. In the speaker change detection step, the system sequentially detects whether a speaker changes in the middle of speech analysis frame assuming no knowledge about the identity and number of speakers.

There are generally two categories of speaker change detection methods: metric-based and model-based. The metric-based method employs the maximum point of an appropriately defined "metric" between neighboring segments for signaling detection. The model-based method on the other hand relies on models for speakers, background noise, speech and music built in advance. The incoming audio stream is then classified, for example, by a maximum likelihood selection over a sliding analysis window [22]. Model-based detection requires both training data and some information about the test data such as the number of speakers. In contrast to the model-based methods, metric-based methods can be executed without such data requirements. However, the former offers potential for progressive adaptation and regression against variability provided data is available.

Unsupervised speaker indexing systems usually use a metric-based method such as the Bayesian information criterion (BIC) or the generalized likelihood ratio (GLR) tests based on the comparison of two statistical models from two adjacent segments. BIC is a likelihood criterion penalized by the model complexity: the number of model parameters [21]. The BIC procedure is to choose the model with which the BIC criterion is maximized. The BIC difference of two competing models can be seen as an approximation to the logarithm of the Bayes factor [23]. BIC has some advantages: robustness and threshold-free. However, its computation is costly. The GLR test is similar to the Bayesian Information Criterion in that it compares two competing models, but it is simpler and less complex to compute. We have adopted the GLR test in our unsupervised speaker indexing algorithm.

To detect speaker changes, we use an analysis window which consists of two segments. The two segments within the window are compared using the GLR test [5], [10]. Suppose there are two feature vector sets, X_1 and X_2, coming from each segment, respectively. Hypothesis, H_0, is that the speakers in two segments are same, while hypothesis, H_1, is that the speakers are different. Let $L(X_1;\lambda_1)$ and $L(X_2;\lambda_2)$ be the likelihood of X_1 and X_2 where λ_1 and λ_2 represent model parameters that maximize each likelihood. Similarly let X be the union of X_1 and X_2. $L(X;\lambda_{1+2})$ is the maximum likelihood estimate for X. Gaussian models are used here, and λ includes the mean and variance of the Gaussian model which are obtained from the data of each segment. Then

$$\mathrm{GLR} = \frac{L(X;\lambda_{1+2})}{L(X_1;\lambda_1)L(X_2;\lambda_2)} \tag{1}$$

When two segments represent the same speaker, GLR value goes up to 1; otherwise, it falls to zero. We apply a preset threshold, which is empirically obtained, on GLR to determine the latent changing point.

B. *Localized Search Algorithm (LSA)*

The length of the analysis segment used for the speaker change detection can be either variable or static. In variable-length segmentation, the speech stream is divided into different lengths depending on several factors such as pauses and background changes. However, static segmentation assumes a fixed analysis segment. A static segmentation is attractive since it is computationally simple but care has to be taken while choosing the analysis segment length. Too short an analysis segment may not provide adequate data for analysis while a longer analysis segment may likely miss a speaker change point. It has been found in our previous experiments that the analysis segments should be at least longer than 2 s for robust recognition [3]. Each analysis segment we considered

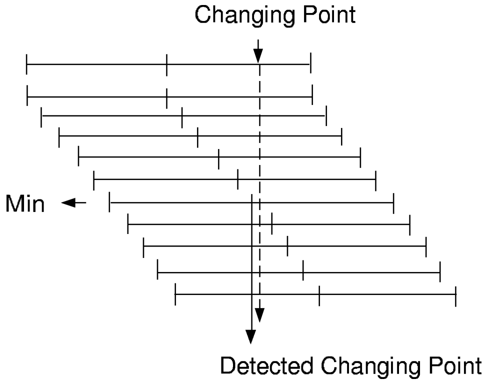

Fig. 2. LSA. Analysis window (4 s) looks for the exact speaker changing point near the potential boundary comparing with two analysis segments (2 s) of the analysis window. The first analysis segment is the reference segment and the segment of the second analysis segment is compared with the reference segment using the GLR test. When the analysis window detects a latent speaker change point, it shifts by 0.2 s to enable a finer search yielding a total of ten ratios from GLR tests. The minimum is chosen, implying the highest probability of a speaker change. The boundary between two analysis segments is recognized as a true speaker changing point.

was 2-s long, and the total analysis window, which includes two adjoining analysis segments, at any point is 4-s long. This implies that we cannot detect a speaker change occurring within an analysis segment shorter than 2 s. Smaller analysis window shifts (e.g., 0.2 s) could lead to finer resolution [5]. But computational complexity severely increases with the number of analysis windows. For example, if the analysis window shifts by 0.2 s, we need ten times more GLR calculations than in the 2-s shift case. To solve this problem, we propose a LSA. Our algorithm seeks a compromise between accuracy and efficiency. Fig. 2 shows an illustration of how this algorithm works.

We assume that the boundary between the two analysis segments is the speaker changing point. The analysis window shifts by 1 s. The analysis window consists of two analysis segments, the first of which is the reference for speaker change detection. The speech data in the second analysis segment are compared with the reference segment to detect whether they are from the same speaker or not. The analysis window, which shifts overlapped by 3 s, is not appropriate to detect the exact speaker changing positions. In other words, we cannot detect a changing point within a 1-s duration with this amount of shift. For that reason, firstly, the analysis window shifts by 1 s. When GLR falls below the threshold, the data in the second analysis segment of the current analysis window may include a latent speaker changing point. Then, the LSA starts running through the 0.2-s analysis window shift for enabling a finer search [5]. There are ten candidates, one of which indicates a true speaker changing point.

IV. Generic Models for Bootstrapping

To build effective speaker models, sufficient training data are required. In the unsupervised scenario, there is no prior knowledge about the speakers. When the speaker indexing process starts, only the data seen thus far can be used for modeling due to the sequential nature of the indexing process. Such models that are constructed roughly on-the-fly can cause severe clustering errors. The key issue here is finding a method for alleviating the model initialization problem. The idea of generic models offers a promising alternative. We can create generic models of speakers that are independent of the test set speakers with the hypothesis that some speakers of the reference set are acoustically close to the test speaker and can be adapted to be closer with new data [18]. Although we do not know the exact number of speakers, we assume that the number is finite. With this assumption, the initial generic models are built through training with data not directly related to the test condition. This can make it possible for the speaker indexing system to operate without training of true speaker models.

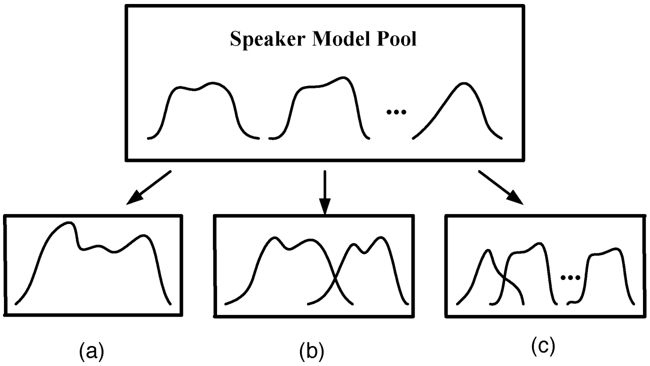

Fig. 3. Generic Models. (a) UBM: the entire speaker data in the pool is used to create a single model. (b) UGM: the data is used to create two gender models. (c) SSM: speaker models are selected from the generic speaker data pool by the proposed sampling method.

There are at least three possibilities that one can consider for creating generic models: UBM, UGM, and the proposed SSM. For example, suppose there are M male speakers and N female speakers in the generic speaker data pool. The UBM is built pooling the entire data of $(M+N)$ speakers. UGM includes two models: one for male speakers (trained with data from M male speakers), and the other for female speakers (trained with N female speakers). SSM is a new generic model set that we have proposed in this paper. At first, we pick S speakers, the number of which is smaller than the total number of speakers in the generic data pool, and construct S speaker models. While UBM and UGM involve "averaging" across a number of speakers, SSM does not [Fig. 3]. When we use SSM as a generic model, we have to address two problems. The first concerns the number of sample models needed, and the other concerns the sampling method needed for constructing the SSM set. At present we lack an analytical way for seeking an optimal choice for these parameters. We evaluate these empirically in each case: for example for the size of SSM, 16 is optimal for two-people conversations, or 32 is optimal for four-people conversations. We assume that this optimal number varies by the number of speakers and the type of speech data including how the features are defined.

The MCMC approach offers a promising approach for model sampling. Monte Carlo methods are computational techniques to make use of random numbers. One of the uses of Monte Carlo method is to generate samples from a given probability distribution. We used the Metropolis algorithm, which is an instantiation of MCMC method. The Metropolis method is widely used for high-dimensional problems [13]–[15]. It generates samples by running an ergodic Markov Chain which converges to a target distribution function f. For an arbitrary starting value $x^{(0)}$, a chain $(X^{(t)})$ is generated using a transition kernel with the stationary distribution f, which ensures the convergence of $(X^{(t)})$ to a random variable from f. Thus, for a "large enough" T_0, $X^{(T_0)}$ can be considered as distributed under f. The number of samples n can be predetermined, and the samples $(X^{(T_0)}, X^{(T_0+1)}, \ldots, X^{(T_0+T_n)})$ are generated from f according to the criterion [16].

In our speaker indexing system, we tried to apply the MCMC method to choose sample speaker models. It can be briefly summed up as follows.

1) The target distribution of the sampling space is estimated from the UBM that represents all the speakers in the pool.
2) The predetermined number of sample vectors are chosen by Metropolis criterion from this initial distribution (a normal distribution is assumed). The mean of this distribution is obtained from the centroid of the UBM.
3) With every sample vector and every speaker model, the likelihood is calculated.
4) A speaker model that provides the maximum likelihood of a sample vector is chosen.

V. Clustering and Model Adaptation

The segments obtained from the speaker change detection step are indexed in terms of speakers, and then the corresponding models are adapted with the newly indexed data. For clustering, we experimented with speaker models from the predetermined generic model sets: UBM, UGM, and SSM. In UBM and UGM cases, the generic pooled model was adapted to create the speaker specific models. Since speaker indexing is a sequential process, the first speaker model is always created from UBM (or UGM) using the first speaker segment. With the next speaker segment, the model just constructed is assumed as a speaker model. However, if the likelihood of the second speaker segment is lower than the threshold, then new speaker model is created by bootstrapping from the UBM (or UGM). The subsequent speaker segments are sequentially clustered in a similar manner: Each time the new data segment is evaluated against all available speaker models. Whenever new speakers are detected, the number of models hence increases in the UBM and UGM case. With SSM, the generic speaker models are adapted into speaker specific models [Fig. 4]. The likelihood of every speaker segment is calculated with the sample speaker models, and the model with maximum likelihood is selected and adapted sequentially. The number of models is kept the same for the entire data [Fig. 5].

Model adaptation is executed by the MAP scheme. As the amount of data increases toward infinity, the MAP estimate converges to the ML estimate [17]. The MAP adaptation on a GMM is straightforward [20]. Given the adaptation vectors $X = \{x_1, x_2, \ldots, x_T\}$, we compute the probability $Pr(i|x_t)$

$$\Pr(i|x_t) = \frac{w_i p_i(x_t)}{\sum_{l=1}^{M} w_l p_l(x_t)} \tag{2}$$

where w_i is the weight of each mixture in the GMM, and p_i is the probability of input, x_t, in each mixture. M is the number of mixtures. In this system, means, $\hat{\mu}$, and weights, $\hat{w}$, of GMM are updated as follows:

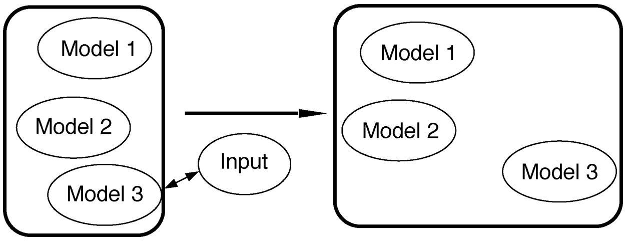

Fig. 4. Model Adaptation: from generic speaker models into speaker specific models.

$$\hat{\mu_i} = \alpha_i^m E_i(x) + (1 - \alpha_i^m)\,\mu_i \tag{3}$$

$$\hat{w}_i = \left[\frac{\alpha_i^p n_i}{T} + (1 - \alpha_i^p)\,w_i\right]\gamma \tag{4}$$

where γ is a scale factor. α_i^m and α_i^p are data-dependent adaptation coefficients which are defined as

$$\alpha_i^m = \alpha_i^p = \frac{n_i}{n_i + r_\rho} \tag{5}$$

where r_ρ is the fixed relevance factor, and the sufficient statistics of mixtures, n_i, and the re-estimation of mixtures, $E_i(x)$, are defined as

$$n_i = \sum_{t=1}^{T} \Pr(i|x_t) \tag{6}$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|x_t) x_t. \tag{7}$$

We assume that speaker models in the reference set are independent of the test speech data. A desirable property of the generic models hence is to ensure rapid adaptation to the true speaker models. Furthermore, the acoustic environment of generic models and test speech stream might be different. If the difference is large, we need to compensate additionally for such effects. To address this problem, for example, we may use the first speaker segment to adjust for the channel difference (e.g., for cepstral mean normalization).

VI. Experiments

We used two audio data sources in this paper: the 1999 Speaker Recognition Benchmark Corpus from NIST (1999) and the HUB-4 Broadcast News Evaluation English Test Material (1999). For the generic model, we used 100 speakers (50 male speakers and 50 female speakers) who were randomly selected from the training data in the NIST Speaker Recognition Benchmark Corpus. For training each speaker model, about one minute of speech data were used. We also tested our speaker indexing system using independent portions of this data corpora. The primary experiment focused on unsupervised on-line speaker indexing. To investigate the convergence performance under multipass ("off-line") conditions, we repeated the indexing over several iterations.

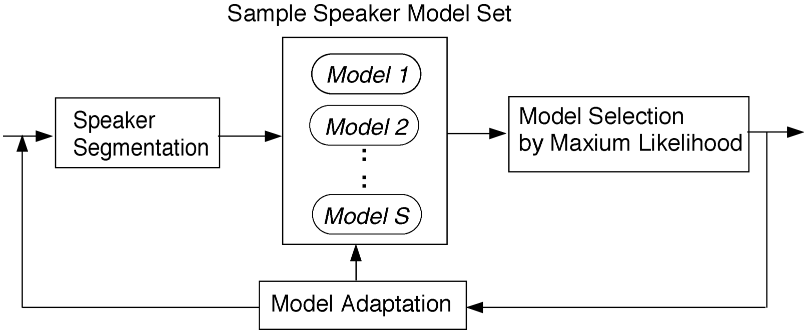

Fig. 5. Clustering with SSM and adaptation.

We performed two main experiments with a variety of speech materials: the first test focused on model adaptation and convergence behavior with various lengths of analysis windows and generic model types while the second test investigated the overall performance with various generic models. Specifically, in the first set of experiments, we evaluated speaker identification error rates with various lengths of analysis windows (i.e., 1, 2, 4, and 8 s) and with the three types of generic models (UBM, UGM, and SSM) to find out the optimal conditions for speaker change detection and model adaptation steps. We randomly picked some speakers from the 100 speaker pool for SSM using the MCMC method. Although the speaker pool consisted of 50 males and 50 females, sampled speakers were not necessarily evenly distributed in gender. The UBM and UGM were also built using data from 100 speakers in the pool. In SSM, we assumed that “16” was an experimentally suboptimal number of speaker models as a reference set to index speakers with the telephone conversation and Broadcast news materials [6].

The other experiment investigated the performance of the generic models. The speaker indexing test included three tests with different speech data sets: two-speaker conversations, four-speaker conversations, and broadcast news. The first test material was executed with about 24-min audio data from the Speaker Recognition Benchmark NIST Speech (1999). The length of each speech audio sequence was about one minute. Each sequence included two-speaker telephone conversations. One third of the sequences included mixed gender (one male and one female) conversations. The other sequences include two males or two female speakers. As for the second test material, about 24-min audio data from the Speaker Recognition Benchmark NIST Speech (1999) were used. Since we needed exactly four-speaker conversations, we created them with two-speaker conversations (“artificial sequences”) of the Speaker Recognition Benchmark NIST Speech (1999). No speaker used for building the generic model participated in the test conversations. One third of the sequences included mixed gender (two males and two females) conversations. The other sequences included four males or four female speakers. The third test material constituted about 45 minute audio data from the HUB-4 Broadcast News Evaluation English Test Material (1999). Broadcast news data included various categories of audio data and environmental conditions. Speaker types in this data included anchors, guests, interviewers, and interviewees. For our experiment, we considered 20 news clips representing different topics, and the number of speakers ranged from two to six speakers. We tested which of the three generic models (UBM, UGM, SSM) showed the best speaker indexing performance on these multiple speaker conversations. In the SSM case, we considered several sample set sizes (i.e., 8, 16, 32, 64, and 100 sample models).

Since long silences have an adverse effect on speaker recognition, we eliminated data segments which were longer than 100 ms and lower than -40 dB as silence. Experimental data were sampled at 8000 Hz. As feature vectors, we used 26-channel, 24-dimensional Mel Cepstrum vectors. We also used a 30-ms Hamming window that was shifted by 10 ms. Speaker models were GMMs with 16 mixtures.

VII. Results

Only the speech portion extracted from the input audio stream was sequentially categorized in terms of distinct speakers using the generic models. The first experiment was relevant to the convergence of model adaptation, and the results are shown in Table I. In this experiment, we wished to determine what length of speaker segment was optimal to index speakers under various conditions such as the number of speakers in the speech sequence and the type of generic models used.

When the length of the segment was increased (e.g., to 8 s) under the same model conditions, speaker indexing error rates decreased to almost 0% in most cases, although in some conditions, the error rates slightly increased (the case with four speakers and UGM). It implies that more speech data in a segment include more discriminatory information to represent a specific speaker. For example, when the number of speakers was 1, in the SSM case, the error rates decreased from 2.2% to 0% as the length of segment increased to 8 s. From our experiments, we found that a segment of 8 s provided the best choice (Table I). No significant improvements were found for segments longer than 8 s. However, we may need a shorter data segment to index shorter speaker segments. For example, to consider a 20-s two-speaker conversation, we used 8-s-long analysis segments. Suppose that one of the speaker’s speech lay between the 5- and 7-s mark in the sequence (as measured from the beginning of the audio stream). Although the first 8-s analysis segment included two speakers, it would be recognized as one speaker. To detect shorter speech episodes, we should use as short an analysis segment as we can. However, shorter (e.g., 1 second) segments could not capture a speaker’s information adequately in

TABLE I
INDEXING ERROR RATES AS A FUNCTION OF AVERAGE LENGTH OF SEGMENT PER SPEAKERS, NUMBER OF SPEAKERS IN THE TEST SEQUENCE, AND GENERIC MODEL TYPES

Average Length of	Generic	Number of Speakers		
Segment per Speakers(sec)	Model Type	1	2	4
1	SSM	2.2%	0%	2.8%
	UGM	3.9%	4.3%	6.6%
	UBM	3.0%	3.0%	5.3%
2	SSM	1.6%	0%	0%
	UGM	1.9%	0%	0.9%
	UBM	2%	0%	0.9%
4	SSM	0.5%	0%	0%
	UGM	0.5%	0%	1.8%
	UBM	0.4%	0%	0%
8	SSM	0%	0%	0%
	UGM	1.0%	0%	3.6%
	UBM	0%	0%	0%

our experiment. We determined empirically from this first experiment, that a 2-s analysis segment offered a good compromise.

As the number of speaker candidates in a conversation increased, the error rates are expected to rise. For example, suppose that we had four speakers in a test sequence. Each speaker had about 1 min of speech data, hence totally about 4 min of data were used for speaker indexing. The point is that the indexing process was executed only sequentially without using any prior target speaker models. While the first minute segments passed through the indexing system, some speaker changing points might be falsely detected. The first adapted speaker model and the generic model(s) were compared with the group of segments that were hypothesized as speech of a certain speaker. After one speaker model was adapted from the generic model, the two models are now compared. Whenever a speaker change occurred, the system looked for the next speaker model. As the number of newly detected speakers increased, the number of models we had to compare increased which in turn might affect the overall error rate. Note that since this corpus consists of fairly clean land-line telephone conversations, there was no significant background noise to adversely affect the recognition. However, in some conditions, the number of candidates did not critically affect the error rate of speaker indexing. This result implies that each speaker spoke for about one minute without speaker changes which gave enough speaker information to adapt speaker models and to discriminate speakers well.

In Table I, we show results with the three types of generic models: SSM, UGM, and UBM. SSM provided the most stable performance across all experimental conditions, both in terms of the number of speaker candidates and the length of the analysis segment. Lower error rates also implied that the concomitant model adaptation was better. Recall that whenever a segment was assigned to a speaker, the corresponding speaker model was updated.

The second experiment focused on speaker indexing on the three different test materials with the three types of generic models. Based on the results of the first experiment, we adopted a 2-s analysis segment length. Fig. 6 shows the unsupervised speaker indexing performance of the generic models for the telephone conversations, four-person conversations, and broadcast news clips. In Fig. 6, the initial UBM was a unitary GMM that was trained with data from all the 100 speakers in the pool. The UGM set consisted of two models: male and female. The SSM set had variable number of models. In our experiments, we used 8, 16, 32, 64, and 100 model sets to find empirically the optimal number of samples for unsupervised speaker indexing under various test conditions.

In the two-speaker case, when the number of generic model speakers was smaller than 16, the indexing accuracy was below 90%. As the number of model samples became larger, the accuracy peaked at about 92.5%, before slowly degrading, as the number of samples increased further. The reason might be that eight models were not adequate to recognize two speakers, as they could not have adequate discriminatory power in our feature space. While the 32-model case (90.1%) performed better than the eight-model case (87.4%), they were still worse than for the 16-model set. However, as the number of models increased further, too many similar models occupied the feature space. In this situation, one (test) speaker could be recognized as two or more (model) speakers. From this experiment, 16 was found to be the optimal number of sample speaker models. The results of UBM (82.8%) and UGM (86.1%) cases were worse than that of the 16-SSM case. Recall that the UBM was built with 100-speaker data, and UGM consisted of a male model and a female model that were built with 50 male and 50 female speaker data, respectively. Each model of SSM, however, was a specific generic speaker model. For that reason, UBM and UGM had larger variances, and each speaker model was adapted from those initial models. Although the model variance could be adapted, it is difficult to represent a speaker well with small amounts of data. In sum, UGM is better than UBM because gender models might have relatively smaller variances.

The results were similar for the four-speaker case; the best performance (89.6% accuracy) was obtained with 16 sample models. When the number of samples was 8, the accuracy was about 69.5%. While for the 32 model case it was 84%. Again, the results of UBM (60.2%) and UGM (67.1%) cases were much worse than for the 16-SSM case. These experiments showed that the automatic selection of initial models in a way most similar to the final target models leads to better and faster model adaptation and convergence.

Based on the results of the two-speaker case and the four-speaker case, the accuracy of two two-speaker indexing was higher than that of four-speaker indexing with all generic models except 100-SSM. The reason might be that the possibility of false indexing increased as the number of speakers to index increases. However, it is interesting that the difference of accuracy decreased as the number of sample models in SSM case increased. And, with 100 sample models, the four-speaker case got slightly higher accuracy than the two-speaker case.

The broadcast news data posed significantly more challenges mainly due to the diversity in the audio accompanying the

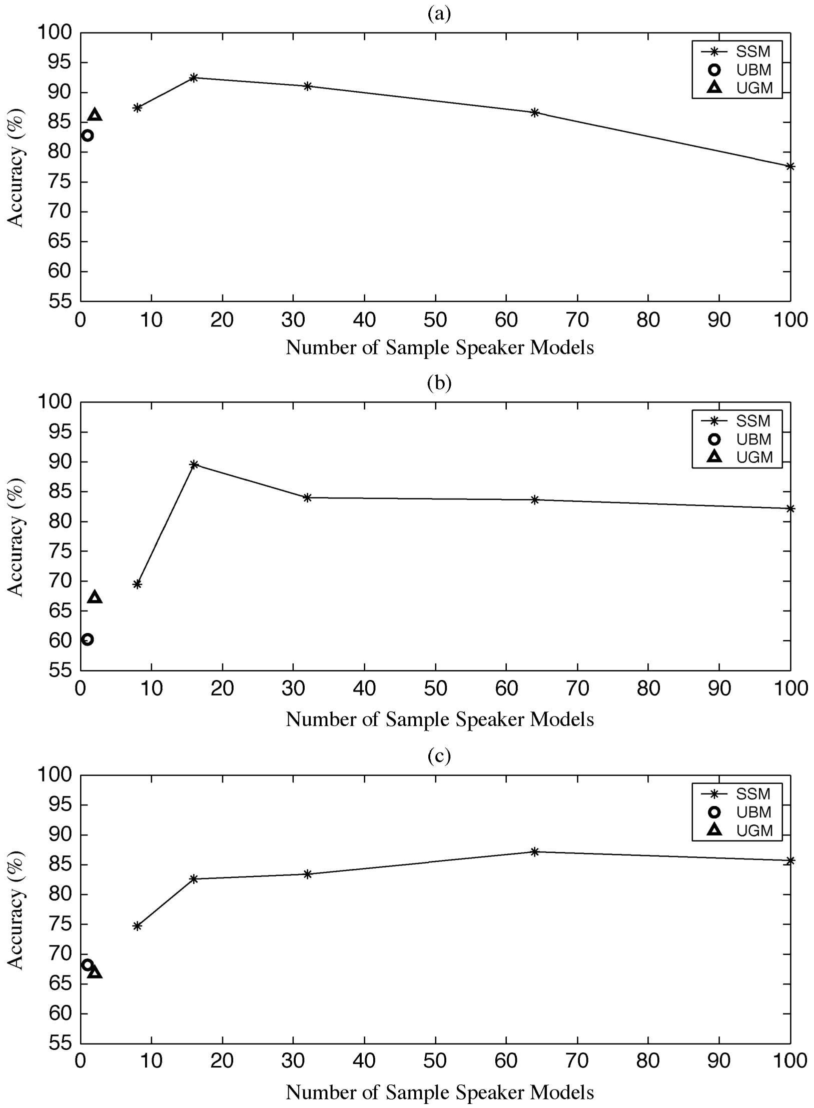

Fig. 6. Indexing accuracy for various types of generic models (SSM, UBM, UGM). (a) Two-speaker conversations. (b) Four-speaker conversations. (c) Broadcast news. Note that "samples" here refer to those drawn from the generic model pool by MCMC. The results for SSM were consistently better than for the UBM and UGM cases.

speech therein. Our test set consisted of 20 audio clips, segmented manually based on topics, from a 45-min broadcast. There were two to sixspeakers in these clips; the distribution of genders was uneven. There were more males than females. Fig. 6(c) shows the speaker indexing result. Even though the best performance was obtained with the sample model set of size 64 and the accuracy was 87.2%, the difference in accuracies with the others (16, 32, 64) was small. There might be several reasons. The different audio data conditions and the variety in the number of speakers (two to six) could affect the results. Based on our analysis, we might need more sample models, but it is not directly proportional to the number of target (true) speakers. The main reason that 64-SSM was best is due to the stability of performance. Even though more sample models make more errors, the variance of error rates in each test was smaller. When the number of speakers is over six (i.e., 10, 16, 32), we might need more sample models, at least equal to the number of target speakers.

Based on this result, we compared the accuracy of 64-SSM with those of UGM and UBM by the number of speakers in the clips (see Table II). With any number of speaker models, SSM with 64 sample speaker models was the best among the three generic models. Even in the six speaker clips, the error rate was 20% which implied the stability of SSM. The performance could however be adversely affected from the environment and individual speakers. That may explain why the accuracy of four-speaker clips were much worse than that of two or five speaker clips in not only the 64-SSM case but also the UGM and UBM cases. In this worse condition, 64-SSM also showed a below 20% error rate for the four-speaker clips.

We also investigated the effect of using clustered speaker models for initialization that uses averaged statistics from a pool of like speakers much similar to universal/gender background models. The main difference from the latter is that the averaged speaker information used for modeling comes from a set speakers deemed to be cohorts under a specific similarity criterion such as the K-L distance. Our hypothesis is that averaging information across speakers would reduce the model discrimination power and that their adaptation into target speaker models in general would be slower. However, if the size of the clusters is

TABLE II
NUMBER OF SPEAKERS IN CLIPS VERSUS ACCURACY FOR THE BROADCAST NEWS MATERIAL

Number of	Length of	Accuracy		
Speakers	clips(sec)	64-SSM	UGM	UBM
2	507.83	93.9%	70.1%	68.8%
3	824.32	89.8%	67.0%	72.0%
4	773.92	80.8%	60.8%	66.2%
5	402.89	89.2%	72.5%	63.6%
6	168.95	80.0%	70.2%	68.8%

small, the results may be comparable to SSM. We did some preliminary experiments to explore the effect of speaker clustering on speaker indexing leveraging recent related work that considers the notion of optimal quantization of the speaker model space (Kwon and Narayanan, 2004). In that work, only one representative speaker was selected from a given quantized region to represent a speaker model. In the experiments for this paper, however, we used the "clusters" generated in the quantization process to create a clustered speaker model corresponding to each quantized region. In our experiments with a portion of the NIST 1999 corpus with 100 speakers, we found that there were 12 clusters and the number of members in each cluster varied from two to 15. The indexing results showed that on average SSM outperformed the clustered version about 4% absolute in unsupervised speaker indexing on two-speaker telephone conversations.

The final experiment investigated the off-line indexing scenario. The condition for the off-line indexing case is different from that for the on-line indexing case since the former case can be processed iteratively in multiple passes through the data (i.e., relax causality constraint). We applied the 16-SSM to the off-line system to see whether it is useful in the off-line environment or not. We used the two- and four-speaker conversations that were used in the on-line speaker indexing test. The result showed that the speaker indexing converged in the first iteration. This may imply two possibilities: The first possibility is that the off-line unsupervised speaker indexing with 16-SSM has the same performance as that of the on-line system. The other is that the selected initial models are adapted and converged well (no changes after the first iteration) in the on-line unsupervised speaker indexing with 16-SSM.

VIII. CONCLUSIONS

We presented a novel method for enabling unsupervised speaker indexing. For an unsupervised sequential process without any prior knowledge about the speakers, a generic model set was incorporated into the general speaker indexing framework. This generic model was shown to help the unsupervised speaker indexing system to overcome some of the difficulties arising due to the lack of data for building true target speaker models. In particular, the SSM approach showed better and more stable performance than the other generic model methods such as UBM and UGM. Since these generic models do not contain the type of speakers training data in the initial for indexing, this implies that we do not have to retrain speaker models whenever we test with different speakers.

We used telephone conversation data and broadcast news to evaluate the performance of our algorithm. The condition that yielded the best performance in our experiments was using 2-s analysis segments in conjunction with 16 sample speaker models for two-speaker conversations. The total error rate was 7.53%, which was about 10% lower absolutely compared with the UBM case (17.21% relative). In the case of four-speaker conversations, 16 sample speaker models performed the best with an error rate, 10.4%, which is about 30% absolutely better than that of UBM. As the number of speakers present in conversations increases, the error rate of UBM increased at a much higher rate than that of 16-SSM model set. In the experiment with the broadcast news, the actual number of speakers was not known. The news clips considered had between two and six speakers. More samples were required than those in the four-speaker telephone conversations to cover the wide range in the number of speakers. The result showed that 64 was the optimal number of speakers for broadcast news clips with an error rate of 12.8%, which is about 20% absolutely better than that of UBM (31.8%). Table II shows that the performance of 64-SSM was stable: the error rate was only 20% in the worst case.

From the experiments in this paper, we can conclude that the SSM approach is more robust and stable to variations in the number of speakers and data types than previously proposed generic model approaches such as the UBM and the UGM. However, there are at least four key issues worth considering to further improve the overall performance of unsupervised speaker indexing: strategies for effectively sampling the SSM set, detecting speaker changes in a robust way, adapting speaker models, and integration of multimodal features.

In this paper, we adopted the MCMC method to pick the samples from the pool. This method attained some measure of success in obtaining suboptimal positions of speaker models in the feature space primarily due to the fact that more samples were picked from the space where more speaker models in the pool were concentrated in the feature space. There are a couple of issues that need further investigation in this context. One critical issue with this SSM approach relates to finding the optimal number of sample models and positions in the feature space to use. For a given feature space, some of the models can be severely overlapped, and some are farther apart, even if this formation can be thought to be inherently natural. A more principled approach, with supporting experiments, is required in organizing the space spanned by the (generic) speakers for SSM, such as feature space or speaker quantization for optimal speaker (model) sampling. Lastly, higher level linguistic information and multimodal features can be integrated to overcome the limitations of the speaker recognition based on just spectral envelope speech features.

REFERENCES

[1] R. J. Mammone, X. Zhang, and R. Ramachandran, "Robust speaker recognition: a feature-based approach," *IEEE Signal Process. Mag.*, vol. 13, no. 5, pp. 58–71, Sept. 1996.

[2] J. P. Campbell, "Speaker recognition: a tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1436–1462, Sep. 1997.

[3] S. Kwon and S. Narayanan, "Speaker change detection using a new weighted distance measure," in *Proc. Int. Conf. Spoken Language Processing*, vol. 4, 2002, pp. 2537–2540.

[4] J.\ Yang, X. Zhu, R. Gross, J. Kominek, Y. Pan, and A. Waibel, "Multimodal people ID for a multimedia meeting browser," in *Proc. 7th ACM Int. Conf. Multimedia, Part 1*, 1999, pp. 159–168.

[5] A. Rosenberg, A. Gorin, and S. Parthasarathy, "Unsupervised speaker segmentation of telephone conversations," in *Proc. Int. Conf. Spoken Language Processing*, vol. 1, 2002, pp. 565–568.

[6] S. Kwon and S. Narayanan, "A method for on-line speaker indexing using generic reference models," in *Proc. Eurospeech 2003*, 2003, pp. 2653–2656.

[7] ——, "A study of generic models for unsupervised on-line speaker indexing," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop 2003*, pp. 423–428.

[8] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmemtation," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 7, pp. 504–516, 2002.

[9] M. Nishida and Y. Ariki, "Speaker indexing for news articles, debates, and drama in broadcasted TV programs," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, vol. 2, 1999, pp. 466–471.

[10] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 1998, pp. 12–15.

[11] T. Wu, L. Lu, K. Chen, and H. Zhang, "UBM-based real-time speaker segmentation for broadcasting news," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2003, pp. 193–196.

[12] D. Liu and F. Kubala, "Online speaker clustering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, 2003, pp. 572–575.

[13] D. J. C. MacKay, "Introduction to Monte Carlo methods," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA: MIT Press, 1999, pp. 175–204.

[14] M. Davy, C. Doncarli, and J. Tourneret, "Supervised classification using MCMC methods," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'2000)*, 2000, pp. 33–36.

[15] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001, pp. 105–114.

[16] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer, 1999, pp. 71–192.

[17] P. C. Woodland, "Speaker adaptation: techniques and challenges," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, Keystone, CO, Dec. 1999, pp. 85–90.

[18] J. Wu and E. Chang, "Cohorts based custom models for rapid speaker and dialect adaptation," in *Proc. Eurospeech*, 2001, pp. 1261–1264.

[19] J.-F. Bonastre, C. Delacourt, T. Fredouille, T. Merlin, and C. Wellekens, "A speaker tracking system based on speaker turn detection for NIST evaluation," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 2000, pp. 1177–1180.

[20] M. Liu, E. Chang, and B.-Q. Dai, "Hierarchical Gaussian mixture model for speaker verification," in *Proc. Int. Conf. Spoken Language Processing*, vol. 2, 2002, pp. 1353–1356.

[21] K. Mori and S. Nakagawa, "Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 413–416.

[22] S. Chen and P. Gopalakrishnan, "Speaker, environment, and channel change detection and clustering via the Bayesian information criterion," in *Proc. DARPA Speech Recognition Workshop*, 1998, pp. 127–132.

[23] B. Zhou and J. H. L. Hansen, "Unsupervised audio stream segmentation and clustering via the Bayesian information criterion," in *Proc. Int. Conf. Spoken Language Processing*, 2000, pp. 714–717.

Soonil Kwon (S'99) received the B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1998 and the M.S. and Ph.D. degrees in electrical engineering from University of Southern California, Los Angeles.

He is currently a senior engineer with Samsung Electronics Co., Ltd. He is also a member of the Signal and Image Processing Institute and the Integrated Media Systems Center, USC. His current research interests include speech signal processing, voice identification and verification, speech and speaker segmentation/classification, and multimodal speaker tracking.

Shrikanth Narayanan (SM'02) received the Ph.D. degree from the University of California, Los Angeles, in 1995.

He was with AT&T Research (originally AT&T Bell Labs), Florham Park, NJ, first as a Senior Member, and later as a Principal member, of its Technical Staff from 1995 to 2000. Currently, he is an Associate Professor of Electrical Engineering, Linguistics, and Computer Science at the University of Southern California (USC). He is a member of the Signal and Image Processing Institute and a Research Area Director of the Integrated Media Systems Center, a National Science Foundation (NSF) Engineering Research Center, at USC. His research interests are in signals and systems modeling with applications to speech, language, multimodal, and biomedical problems. He has published over 130 papers and holds three U.S. patents.

Dr. Narayanan was an Associate Editor of the IEEE TRANSACTIONS OF SPEECH AND AUDIO PROCESSING from 2000 to 2004 and serves on the Speech Communication Technical Committee of the Acoustical Society of America. He is a member of Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu and received an NSF CAREER award, a USC Engineering Junior Research Award, and a faculty fellowship from the USC Center for Interdisciplinary Research.