Digital Signal Processing **10**, 93–112 (2000)
doi:10.1006/dspr.1999.0359, available online at http://www.idealibrary.com on IDEAL®

Approaches to Speaker Detection and Tracking in Conversational Speech [1]

Robert B. Dunn, Douglas A. Reynolds, and Thomas F. Quatieri

M.I.T. Lincoln Laboratory, 244 Wood St., Lexington, Massachusetts 02420

E-mail: rbd@sst.ll.mit.edu, dar@sst.ll.mit.edu, tfq@sst.ll.mit.edu

Dunn, Robert B., Reynolds, Douglas A., and Quatieri, Thomas F., Approaches to Speaker Detection and Tracking in Conversational Speech, *Digital Signal Processing* **10** (2000), 93–112.

Two approaches to detecting and tracking speakers in multispeaker audio are described. Both approaches use an adapted Gaussian mixture model, universal background model (GMM-UBM) speaker detection system as the core speaker recognition engine. In one approach, the individual log-likelihood ratio scores, which are produced on a frame-by-frame basis by the GMM-UBM system, are used to first partition the speech file into speaker homogenous regions and then to create scores for these regions. We refer to this approach as *internal segmentation*. Another approach uses an *external segmentation* algorithm, based on blind clustering, to partition the speech file into speaker homogenous regions. The adapted GMM-UBM system then scores each of these regions as in the single-speaker recognition case. We show that the external segmentation system outperforms the internal segmentation system for both detection and tracking. In addition, we show how different components of the detection and tracking algorithms contribute to the overall system performance.

Key Words: speaker recognition; detection; tracking; multispeaker; Gaussian mixture model; clustering

1. INTRODUCTION

With the increasing availability of archived audio material comes an increasing need for efficient and effective means of searching and indexing through this voluminous material. Searching or tagging speech based on who is speaking is

[1] This work was sponsored by the Department of Defense under Air Force Contract F19628-95-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Air Force.

1051-2004/00 $35.00

one of the more basic components required for dealing with audio archives, such as recorded meetings or the audio portion of broadcast shows. Traditional approaches to speaker recognition, however, are designed to identify or verify the speaker in a speech sample known to be spoken by a single person. For audio indexing or searching, the basic recognition approach needs to be expanded to handle both detection and tracking of speakers in multispeaker audio. In this paper, we present two approaches for developing such multispeaker detection and tracking systems.

The systems described below were developed for the multispeaker detection and tracking spokes of the 1999 NIST speaker recognition evaluation [1]. The data for these tasks consist of two person, conversational telephone speech from the Switchboard-II corpus. Unlike Broadcast News audio, these data do not explicitly contain nonspeech events like music, but present other challenges such as handset variability. Given an audio file containing conversational speech and given a hypothesized speaker, the task of detection is to determine if the hypothesized speaker is talking in the audio file. This task is the same as the traditional single-speaker detection or verification task except there is no prior knowledge that the audio file contains speech from only one person. The tracking task is to determine where in the audio file, if at all, the hypothesized speaker is talking[2]. In both cases, performance is computed in terms of the detection errors, misses and false alarms, and presented via detection error tradeoff (DET) plots [3]. Details of the 1999 NIST evaluation data and metrics can be found in [1].

In a canonical single-speaker detection system, a likelihood ratio statistic between a model of the hypothesized speaker and a background model representing the alternative hypothesis is computed using all speech in an audio file since it is assumed that all the speech was produced by a single speaker. When the audio file contains speech from more than one speaker, a likelihood ratio statistic produced using all the speech is contaminated and is unreliable for accurate decision making. An obvious approach to the analysis of multispeaker speech is to segment the speech stream into speaker homogeneous segments and then obtain likelihood ratio scores over these single-speaker segments: in effect, turn the multispeaker problem into a sequence of single-speaker problems. The segmentation of the speech into speaker homogeneous regions can be accomplished in two ways. The *internal segmentation* approach uses a sequence of time-varying values of a running likelihood ratio statistic computed over short segments of speech to determine regions most likely to have been produced by the hypothesized speaker. In the *external segmentation* approach, a segmenter, which does not use knowledge of the hypothesized speaker, is used to produce speaker homogeneous regions, generally by some form of sequential speaker change statistic and/or blind source clustering of short speech segments. Likelihood ratios are then produced over these putative single-speaker regions

[2] The more general task of tracking speakers in an audio cut with no prior hypothesized speaker information is not currently part of the NIST evaluations but has been addressed in several speech recognition systems applied to the DARPA Broadcast News task [2]. The primary goal in these DARPA systems is tracking and clustering for the adaptation of speech recognition models.

for detection or tracking. In this paper we present systems which employ both internal and external segmentation for the multispeaker detection and tracking tasks.

The Gaussian mixture model, universal background model (GMM-UBM) speaker detection system developed at MIT Lincoln Laboratory [4, 5] is used to compute the likelihood ratio which is central to both the detection and tracking tasks. The GMM-UBM system is a likelihood ratio detector consisting of a large, speaker-independent GMM representing the alternative hypothesis (i.e., the UBM) and an adapted GMM representing the hypothesized speaker. This adapted GMM is derived from the UBM via Bayesian adaptation using training data. The GMM-UBM system is used as the likelihood ratio score generator for the detection and tracking systems because it performs single-speaker detection with high accuracy and because it imposes no temporal constraints on input segment size. The system can therefore generate scores both for very short speech segments and for agglomerations of segments which may be collected from scattered locations throughout a speech file.

The remainder of the paper is organized as follows. In Section 2, we describe in more detail the basic front end processing, including features and channel compensation, and models of the GMM-UBM system used for the likelihood ratio computation in the detection and tracking systems. Our internal and external segmentation systems for detection and tracking are then described in Sections 3 and 4, respectively. In Sections 5 and 6 we present experiments and results on the NIST 1999 multispeaker recognition evaluation using our detection and tracking systems. Finally, discussion of results and conclusions are given in Sections 7 and 8.

2. FRONT END PROCESSING AND MODELING

The GMM-UBM system is essentially a likelihood ratio detector consisting of front end processing to extract features from the input speech and compensate for linear channel effects, followed by computation of the likelihood of these features against models of the hypothesized speaker and a speaker-independent alternative (see Fig. 1). The ratio (or difference in the log domain) of the hypothesized and alternative model likelihoods is the likelihood ratio. In addition, the likelihood ratio score can be further processed to normalize for speaker and handset biases, such as by using HNORM [6].

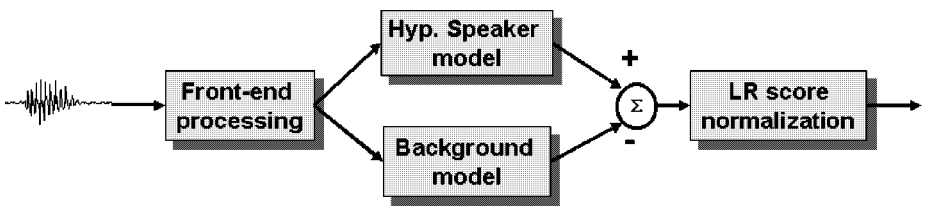

FIG. 1. GMM-UBM likelihood ratio detector.

The front end processing consists of three main steps: feature vector extraction, speech detection, and channel compensation. The features and compensation used in the front end processing were designed to operate on telephone speech. Feature vectors are composed of 19 mel-cepstra and 19 delta cepstra. These vectors are computed every 10 ms by windowing the input speech with a 20 ms Hamming window, computing the log magnitude FFT, and processing that through a 24 filter mel-filterbank. The 24 filters cover the 4 kHz of the signal. The cepstra are then computed from the output of those mel filters which cover the speech band of the typical telephone channel, 300–3300 Hz. The zeroth cepstral coefficient is discarded, and finally, delta cepstra are computed using a first order orthogonal polynomial fit over ± 2 feature vectors from the current vector [7].

Speech activity is detected using an adaptive energy-based speech detector [8]. This detector tracks the noise energy floor of the input signal and declares as speech any feature vector with energy that exceeds the current noise floor by a fixed energy increment. For Switchboard-type telephone speech, it removes about 20–25% of the signal from conversational speech.

In the single-speaker GMM-UBM system, linear channel normalization is achieved with either cepstral mean subtraction (CMS) or RASTA processing [9]. When there is only one speaker present in the speech, hence only one channel characteristic, both of these methods have comparable performance, but in the multispeaker case each speaker potentially has his or her own channel characteristic. In multispeaker speech the mean values of the cepstral coefficients computed over the entire audio file no longer provide an estimate of the channel spectrum so a time-adaptive method of channel normalization, such as RASTA processing, should be used. In fact, application of CMS can distort the features since the (weighted) averaged long-term spectra of both speakers will be subtracted from the features creating a new "channel" effect. We observed a 12.5% decrease in equal error rate (EER) in multispeaker detection when using RASTA instead of CMS.

Both the hypothesized speaker and the alternative model are represented by Gaussian mixture models. The alternative model is referred to as a UBM and is trained using speech from a large number of speakers to create a speaker-independent representation of the distribution of the feature vectors. The speech used to create the UBM should match the characteristics of the speech to be rejected during recognition. In single-speaker detection, where it is commonly assumed that the gender of the unwanted imposter speakers and the hypothesized speaker are the same, gender-dependent UBMs matching the gender of the hypothesized speaker are typically used. To operate on multispeaker audio, however, a gender-independent UBM is used since there is no control over the gender of the competing speakers. In the system used in this paper, a speaker- and gender-independent 2048-mixture UBM was constructed as follows. First, two 1024-mixture, speaker-independent, gender-dependent GMMs were trained, each using 60 min of speech selected from the 30-s tests comprising the 1997 NIST evaluation. These 1024-mixture GMMs were then

combined to form a gender-independent 2048-mixture GMM by agglomerating the mixture components and renormalizing the mixture weights.

Given a UBM, speaker models are then derived using Bayesian adaptation. Using the 2 min of training speech given for the speaker, a one pass, data-dependent adaptation of the parameters of the UBM is used to derive the speaker model. For the systems used in this paper, only the means of the mixture components are adapted. This adaptation essentially adjusts the speaker-independent feature distribution to match the speaker-dependent feature distribution observed in the training data. Details of the adaptation equations can be found in [4, 5].

3. INTERNAL SEGMENTATION

In the internal segmentation approach to multispeaker detection and tracking, a time-varying likelihood ratio score produced by the core GMM-UBM system is used to both segment the multispeaker audio and produce a final score. Given a sequence of feature vectors extracted from an audio file, $\{x_1, x_2, \ldots, x_T\}$, the GMM-UBM system produces a per-vector log-likelihood ratio,

$$LLR[t] = \log(L_{\text{hyp}}[t]) - \log(L_{\text{ubm}}[t]), \tag{1}$$

where $L_{\text{hyp}}[t]$ is the likelihood from the hypothesized speaker model and $L_{\text{ubm}}[t]$ is the likelihood from the UBM for feature vector x_t. Each element of $LLR[t]$ is computed from a single feature vector so the function $LLR[t]$ is very noisy and must be smoothed before it can be used for segmentation. The internal segmentation systems operate by smoothing the time-varying log-likelihood ratios and using this smoothed version to segment the input speech into regions likely to contain the hypothesized speaker. As in the single-speaker detection case, handset variability between training and testing data can cause considerable errors in the likelihood ratio scores [5, 6], so a form of handset normalization (HNORM) is applied to $LLR[t]$ to help alleviate this problem.

3.1. Handset Type Estimation and HNORM

The direct application of HNORM to multispeaker speech is problematic. In the single-speaker case, a handset detector computes the putative handset label for a segment of speech. The appropriate HNORM parameters for a hypothesized speaker model are then applied to the log-likelihood ratio score for the speech segment. In multispeaker speech, it is not appropriate to assume a single handset label for the entire audio file so we must use a time-varying method for applying HNORM. Since the internal segmentation approach relies on the log-likelihood scores to perform segmentation, it is important to apply HNORM to the time-varying function $LLR[t]$ prior to segmentation. In the external segmentation approach discussed later, the speech is presegmented and HNORM can be applied to individual or agglomerated segments as in the single-speaker case.

The time-varying HNORM is applied as follows. On a sequence of feature vectors extracted from the audio file after speech detection, a per-vector likelihood is computed against a GMM of carbon-button transduced speech, $L_{\text{carb}}[t]$, and electret transduced speech, $L_{\text{elec}}[t]$. The GMMs for carbon-button and electret speech are trained using speech from the Lincoln Laboratory Handset Database (LLHDB) [6]. Then, under the hypothesis that carbon-button and electret microphones are equally probable, we compute the per-vector posterior probability of carbon-button as

$$P_{\text{carb}}[t] = \frac{\prod_{\tau=-T/2}^{T/2} L_{\text{carb}}[t+\tau]}{\prod_{\tau=-T/2}^{T/2} L_{\text{carb}}[t+\tau] + \prod_{\tau=-T/2}^{T/2} L_{\text{elec}}[t+\tau]}. \tag{2}$$

The value of T should be large enough to adequately smooth the noisy likelihood functions but small enough to provide good time resolution for detecting changes in handset labels. We have observed that a value of $T = 300$ (corresponding to 3 s) gives reasonable results.

Time-varying handset labels are then obtained by applying a threshold to $P_{\text{carb}}[t]$,

$$HS[t] = \begin{cases} \text{CARB}, & P_{\text{carb}}[t] \geq \theta_{\text{hs}} \\ \text{ELEC}, & P_{\text{carb}}[t] < \theta_{\text{hs}}, \end{cases} \tag{3}$$

where a value of $\theta_{\text{hs}} = 0.6$ was used in the systems described herein. Finally, $HS[t]$ is passed through a 201 point (2 s) median filter to impose constraints on switches between handset labels.

For a hypothesized speaker model, HNORM means and variances are computed for electret and carbon-button speech using handset-labeled 3-s segments from the 1997 NIST evaluation. HNORM scores on a multispeaker audio file are then

$$LLR^{\text{HNORM}}[t] = \frac{LLR[t] - \mu(HS[t])}{\sigma(HS[t])}. \tag{4}$$

To minimize notation clutter, we will drop the HNORM designation on $LLR[t]$ when it is clear that we are using HNORM.

3.2. Speaker Detection Using Internal Segmentation

Our approach to internal segmentation for the speaker detection task is to use the time-varying log-likelihoods to select regions where the hypothesized speaker most likely is located and use these regions to produce a detection score for the entire audio file. The HNORMed log-likelihood function, $LLR[t]$, however, is still a noisy function which must be smoothed to extract useful segmentation information. For detection, we smooth this function using a 101 point boxcar filter, $h[t]$,

$$LLR_{\text{sm}}[t] = LLR[t] * h[t], \tag{5}$$

where $*$ is the convolution operator. Note that for the detection system, $LLR[t]$ is computed only over feature vectors which passed the speech detector. Thus not all time in the audio file is accounted for in the detection system.

Regions most likely to contain the hypothesized speaker are then obtained by applying a threshold to $LLR_{\text{sm}}[t]$,

$$DET[t] = \begin{cases} \text{HYP}, & LLR_{\text{sm}}[t] \geq \theta_{\text{det}} \\ \text{BKG}, & LLR_{\text{sm}}[t] < \theta_{\text{det}}. \end{cases} \tag{6}$$

The threshold θ_{det} is a data-dependent threshold set such that 20% of $LLR_{\text{sm}}[t]$ in the audio file is above the threshold (80th percentile of the distribution of $LLR_{\text{sm}}[t]$ values). The value of 20% was chosen because it gave the best performance on development data. The function $DET[t]$ is further processed with a 101 point median filter to remove unrealistically frequent decision switches.

The final detection score for the audio file is computed as the average of the smoothed log-likelihood ratio function over all regions detected as coming from the hypothesized speaker,

$$S = \frac{1}{|\{t : DET[t] = \text{HYP}\}|} \sum_{\{t : DET[t] = \text{HYP}\}} LLR_{\text{sm}}[t]. \tag{7}$$

Note that averaging the smoothed log-likelihood values instead of the unsmoothed log-likelihood values has the effect of deemphasizing values at detected segment boundaries and has been observed to improve performance [3].

The above approach is related to a previously published approach to speaker verification using multispeaker speech in [10]. In [10], likelihood ratio scores were computed over nonoverlapping, fixed length segments and a detection score was computed either by averaging the top N segment scores or all segments scores which passed a fixed threshold (clip scoring). The above internal segmentation detection system is a generalization of this approach. The system in [10] can be derived from the above system by removing HNORM and decimating $LLR_{\text{sm}}[t]$ by the duration of the boxcar filter.

3.3. Speaker Tracking Using Internal Segmentation

The tracking system is very similar to the detection system but with some modifications. In the detection system it was sufficient to use only the regions most likely to include the hypothesized speaker because a single detection score was required for the entire audio file. This also means that feature vectors detected as silence by the energy detector could be discarded prior to further processing as there is little concern for removing low-energy speech regions. The tracking system, however, must account for the presence or absence of the hypothesized speaker throughout the entire audio file. In this case it is necessary to be more careful about discarding low-energy speech regions as

[3] This occurs because the values in $LLR_{\text{sm}}[t]$ are computed using overlapping windows of $LLR[t]$.

silence and to use the entire function $LLR_{\text{sm}}[t]$, not just a small subset of it. In addition, HNORM is not used in the internal segmentation tracking system as it did not improve performance. One likely explanation for this is that, unlike the detection system which averages over a potentially large set of segments in Eq. (7), the tracking system must report scores over short intervals and HNORM is known to be less effective for short duration segments of speech.

In development testing we found that there were areas in the audio file that our speech detector labeled as silence but the answer keys labeled as speech. This resulted in a minimum miss rate of around 10%. Rather than tuning our speech detector to match the speech detection of the answer keys (which were machine generated and subject to change), we instead compute $LLR[t]$ over all vectors and let the log-likelihood values account for silence regions.

For the tracking system, $LLR_{\text{sm}}[t]$ is computed using a 251 point (2.5 s) triangular filter. Empirically, this triangular filter gave better performance than the shorter boxcar filter used in the detection system. Detected regions are determined as in Eq. (6) using a threshold, θ_{det}, to detect 40% of the vectors as belonging to the hypothesized speaker (and 60% as not) and smoothing $DET[t]$ with a 101 point (1 s) median filter. The detection system uses a higher value for θ_{det} than the tracking system uses because the detection system scores only the region most likely to contain the hypothesized speaker and it ignores the rest of the audio file. The tracking system, on the other hand, must score all regions of the audio file. In addition, the cost function used in the NIST evaluation was optimized by operating the system with a 1–5% false alarm rate, and using development data we found that setting θ_{det} to detect 40% of the data gave the lowest miss rate in this region. For each detected segment, temporally connected regions with the same detection label, $LLR_{\text{sm}}[t]$, is averaged over the segment and that average score is reported for the whole segment. This internal segmentation tracking system is similar to the approach presented in [11].

As with the detection system, we can also simply use fixed-length segment scoring for tracking. For this case, the smoothed log-likelihood ratio function, $LLR_{\text{sm}}[t]$, can be decimated and tracking scores reported at regular fixed intervals. We add a small negative bias to the score of segments which are centered on a detected silence vector. Empirically it was found that using a triangular or Hamming filter of 251 points (2.5 s) for smoothing $LLR[t]$ and reporting scores every 25 vectors (0.25 s) gave the best performance (decimation was required to limit the size of scoring files sent to NIST). No single filter duration gave the best performance at all DET points, rather different durations give the best performance for different points on the DET curve. As shown in the experiments section, the fixed segment approach had better performance than internal segmentation on the tracking task and was used as our primary system for the 1999 NIST evaluation.

It should be noted, however, that in a practical application of a tracking system one would almost always need to select temporally connected regions of speech from the sequence of fixed segment scores, thus using a system more like the first internal segmentation tracking system. The better performance of the fixed-segment tracking system can be attributed to the fact that no hard

decisions of regions or production of a single score for a region was required. This is, perhaps, a flaw in the scoring mechanism for the tracking task.

4. EXTERNAL SEGMENTATION

In the external segmentation approach the audio file is first segmented into speaker homogeneous regions by an independent process before computing log-likelihood values for detection or tracking. In this paper we use a blind clustering approach described in [12] to generate homogeneous regions with no prior knowledge of the hypothesized speaker. For speaker detection, we score each homogeneous region as in the single-speaker case and then take the maximum score as the overall detection score. For speaker tracking, the log-likelihood value of the hypothesized speaker is computed for each region and reported with the region's segmentation times.

The external segmenter used in this paper is a hierarchical agglomerative clustering system which works as follows [12]. The audio file is processed to produce 23 dimensional mel-cepstra feature vectors with no delta coefficients and no channel compensation. Feature vectors from silence regions are removed. We use different front-end processing for the external segmenter than for standard speaker detection because we want to take advantage of channel differences between the speakers to aid in segmentation. The sequence of remaining feature vectors is first partitioned into equal length segments (typically 100 vectors or 1 s). These segments form the initial set of clusters, each containing only one segment. Agglomerative clustering then proceeds by computing the pairwise distance between all clusters and merging the two clusters with the minimum distance. This is repeated until the desired number of clusters is obtained.

The pairwise distance between clusters is based on the likelihood ratio between the likelihood the segments in the two clusters were generated by two different speakers and the likelihood the segments in the two clusters were generated by the same speaker [13]. As introduced in [12], these likelihoods are computed using tied GMM density functions. For each segment, mixture weights to a common, fixed set of Gaussians are estimated. In these experiments, we use a set of 64 Gaussians trained using the entire sequence of feature vectors from the file being segmented. The use of tied GMMs provides better density modeling for the segments than the standard approach of using a unimodal Gaussian density. When two clusters are merged, new mixture weights using the union of segments in both clusters are estimated and distances to the remaining clusters are recomputed. Complete details of this approach can be found in [12].

The output of the clustering is a collection of speaker-homogeneous regions in the original multispeaker speech associated with each cluster produced. Since this is a blind clustering approach, there is no guarantee that the final clusters will represent different speakers, but the relatively long initial segments and uncompensated channel differences between the speakers tend to bias the clustering away from converging on phonetic similarity. Initial testing on two-

speaker Switchboard speech found the clusters produced are 90% pure on average.

For the NIST multispeaker speech it is known *a priori* that there are only two speakers in the audio file. Thus the difficult task of determining the number of speakers is not addressed. However, it is believed that the clustering approach used will work well even with only a general idea of the number of expected speakers since it was found that over clustering (in this case, using three to six clusters) does not adversely affect performance for detection or tracking and can actually provide better performance than exactly matching the number of speakers in the audio file in some cases. There are, of course, several other techniques possible for external segmentation [14, 15] which attempt to detect the number of speakers present.

4.1. Speaker Detection Using External Segmentation

Once the audio file has been clustered, the speech associated with each cluster is scored as in the single speaker case. The values of $LLR[t]$ are averaged across the cluster, the handset label is estimated for the cluster, and HNORM is applied. The maximum hypothesized speaker score of all the clusters is used as the detection score for the entire audio file. This process is shown in Fig. 2. Although as the number of clusters increases, the chances of a spurious high score of a cluster from a audio file not containing the hypothesized speaker increases, we found that in practice a small increase in the number of clusters did not significantly affect performance.

4.2. Speaker Tracking Using External Segmentation

In the speaker tracking problem the output of the external segmenter can be used in one of several ways. The segmenter generates homogeneous regions of speech such as regions 1, 2, and 3 in Fig. 3. One method of speaker tracking is to compute scores across each of the three regions and to use that single regional score for all time locations within the region. A second method is to individually score the segments which compose the different regions (a, b, c, ... in Fig. 3).

In the second approach when using the smaller segments (a, b, c, ...) the length of these segments can vary a great deal. The mean of the segment score is normalized by dividing the score by the segment length (i.e., taking the average score), but there is the additional problem that the variance of scores generated from short segments will be much larger than the variance

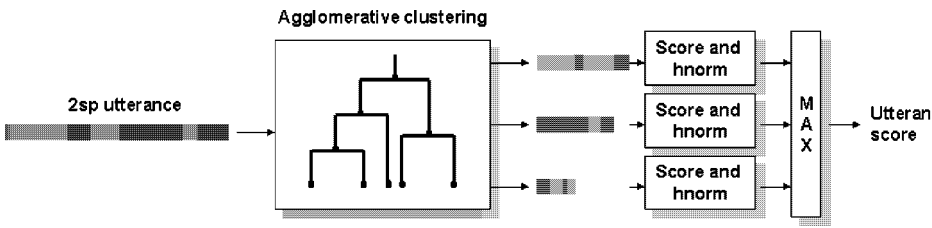

FIG. 2. Speaker detection using external segmentation.

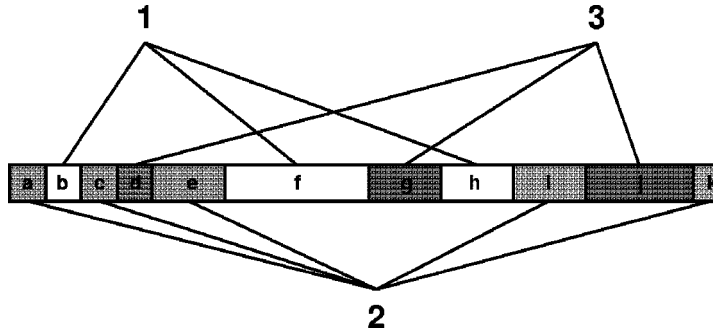

FIG. 3. An example of homogenous regions generated by automatic clustering.

of the scores generated from longer segments. To account for this difference in variance, we measured the variance of nontarget scores as a function of segment length on development data. We then normalize both the variance and mean of the segment score as a function of segment length. This normalization gives a significant improvement in some parts of the DET curve. In particular, it improves performance in the low false alarm rate region (for false alarm rates less than 10%) and in the high false alarm rate region (for false alarm rates greater than 80%). The length normalization has a negligible effect near the EER point.

The best overall performance was from the first method in which one score was computed over each region. This is not surprising because the performance of speaker recognition systems improves when the test segment duration is increased. The first method computes scores over relatively long segments, while in the second method scores are computed over segments as short as 1 or 2 s.

In the tracking task we must also reinsert the silence regions when reporting the final scores. As when tracking with an internal segmentation system, if we give silence an arbitrary low score then there is a floor in the miss rate, in this case around 7%. We handle this problem by scoring the silence region with nearly the same method as the other regions. This requires estimating the handset label during silence where the meaning of the estimate is questionable because the GMMs for the handsets detector were trained with the silence regions omitted. Nevertheless, using the handset estimate for the silence region and applying HNORM does improve the overall system performance. The silence regions tend to be shorter than the other regions so the scores generated for silence regions have a higher variance than scores for the other regions. This problem is addressed by clipping the silence scores to a value of 1.0. That is, if the score during silence is greater than 1.0, the score is reset to 1.0.

5. SPEAKER DETECTION EXPERIMENTS

This section describes speaker detection experiments performed on multi-speaker audio using both internal and external segmentation. The data set used

is that of the two speaker detection task in the 1999 NIST evaluation [1]. The data consist of conversational telephone speech with 1723 test conversations that are each nominally 1 min in duration. There are 2 min of training data for each hypothesized speaker. In our experiments, we present results based on pooling scores from all 1723 test conversations.

The speaker detection system using internal segmentation described in Section 3.2 was the primary system submitted by MIT Lincoln Laboratory in the two speaker detection task of the 1999 NIST evaluation. The performance of the system is shown in the DET plot in Fig. 4, where the dashed line denotes the system performance without HNORM and the solid line shows the performance with HNORM. The use of HNORM reduces the EER from 19.2 to 16.8%.

The statistical significance of the results in Fig. 4 are shown by plotting a rectangle around the EER point of each curve indicating the 90% confidence interval. This rectangle is computed under the assumption that each detection test is an independent trial and that misses and false alarms are decorrelated errors. The 90% confidence rectangle at the operating point $(P_{\text{miss}}, P_{\text{fa}})$ is bounded by the values

$$P_{\text{miss}} \pm 1.645\sqrt{\frac{P_{\text{miss}}(1-P_{\text{miss}})}{N_{\text{tgt}}}} \quad \text{and} \quad P_{\text{fa}} \pm 1.645\sqrt{\frac{P_{\text{fa}}(1-P_{\text{fa}})}{N_{\text{imp}}}}, \tag{8}$$

where P_{miss} is the probability of miss, P_{fa} is the probability of false alarm, N_{tgt} is the number of target trials (3158), and N_{imp} is the number of imposter trials (34,748). The confidence bound is tighter along the false alarm axis because there are roughly ten times the number of imposter trials as there are

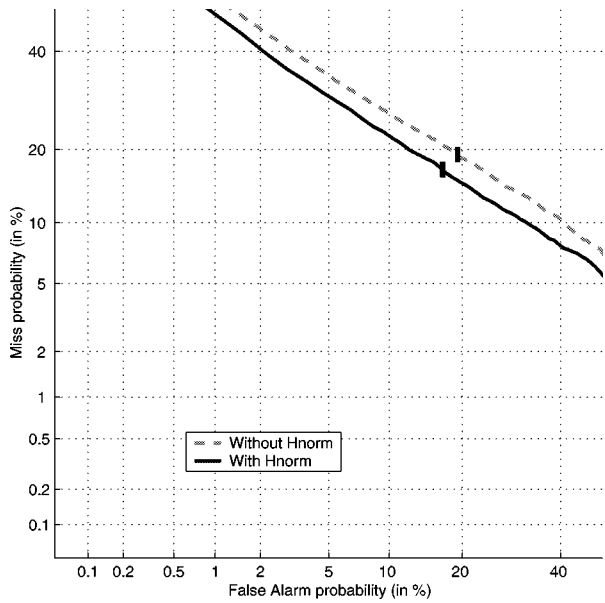

FIG. 4. Two-speaker detection using internal segmentation. The solid line is the performance of the system with HNORM and the dashed line is the performance without HNORM.

target trials. The nonoverlapping error rectangles indicate that the performance improvement from the application of HNORM is statistically significant.

The external segmentation system was developed too late for submission in the 1999 NIST evaluation and is thus not an official submission, but in comparison to our primary submission it has superior performance. Two important parameters that must be set for the external segmenter are the initial segment size and the number of clusters into which the segments are grouped. We examined initial segment sizes ranging from 0.25 s to 1.25 s and found that the performance was not very sensitive to the segment size. We also varied the number of clusters from two to six and found that the performance varied negligibly when using between two and four clusters, although for five and six clusters the performance was slightly reduced. We then chose 1 s as the initial segment duration and three as the number of clusters. The use of three clusters on two speaker speech has been observed to help with lopsided conversations.

Figure 5 shows the performance of the external segmentation detection system with and without HNORM. The use of HNORM is seen to reduce the EER from 17.5 to 15.3%. As in the previous DET for internal segmentation, we show the 90% confidence rectangle around the EER point, indicating again that the performance improvement is statistically significant.

The performance of the internal and external segmentation systems are compared in Fig. 6. The external segmentation system with HNORM, which is plotted with the thick solid line, outperforms the internal segmentation system with HNORM, which is plotted with the thick dashed line. The EER of the external segmentation system is reduced over the EER of the internal segmentation system from 16.8 to 15.3%.

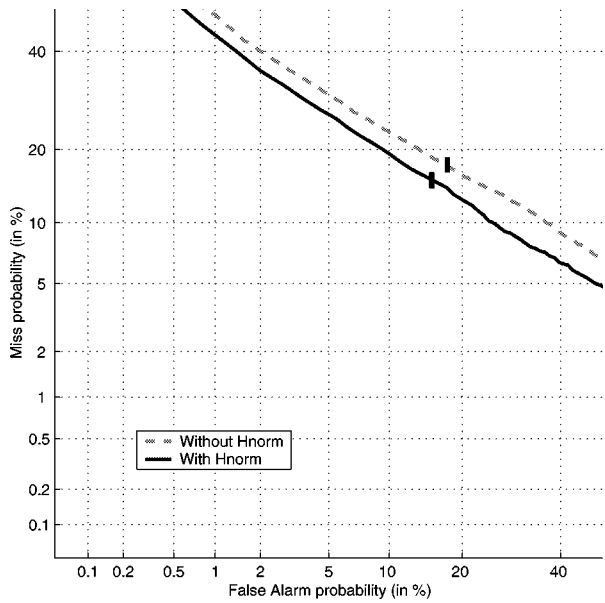

FIG. 5. Two-speaker detection using external segmentation. The solid line is the performance of the system with HNORM and the dashed line is the performance without HNORM.

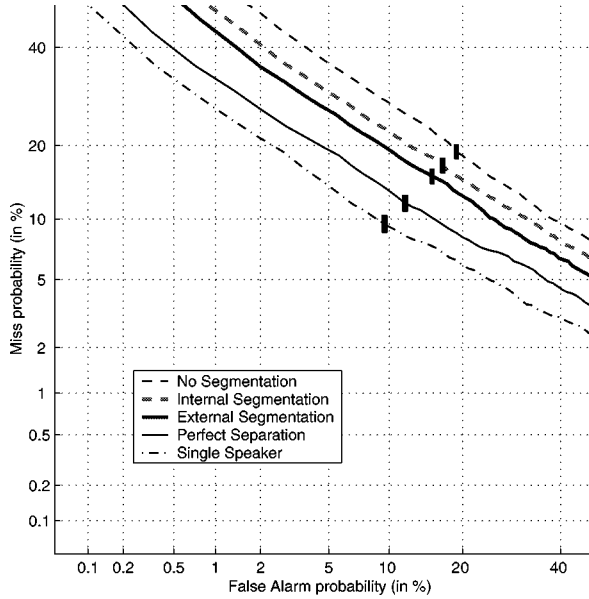

FIG. 6. Comparison of two-speaker detection systems. The thin dashed line shows DET performance for the system with no segmentation, the thick dashed line for the internal segmentation system, the thick solid line for the external segmentation system, the thin solid line for the perfect separation system, and the thin dashed-dot line for the single speaker detection system operating on the individual sides of the two-speaker conversations.

The DET in Fig. 6 also contrasts the performance of these systems with lower and upper bounds of performance. The lower bound (upper right plot) shown as the thin dashed line, corresponds to no segmentation where all speech is scored as if in the single-speaker case. Even with no segmentation, HNORM is applied by estimating a single handset likelihood from all speech frames[4]. Using HNORM with no segmentation gives a uniform improvement in the DET curve and reduces the EER from 20.2 to 19.0%. While comparison to the internal and external segmentation system DETs does indeed show these system impart a large improvement in performance, it is interesting to note that the performance in this worst-case scenario is not as poor as might be expected.

The upper bound of performance, shown as the thin solid line in Fig. 6, is the case when using perfect separation of the two speakers. This perfect separation is generated by scoring each side of the multispeaker conversation separately as a single-speaker test and then taking the maximum score as the overall detection score. The individual sides of the multispeaker conversations are obtained by using the appropriate test files from the one-speaker evaluation. Comparing the perfect separation system to the internal and external segmentation systems shows the loss attributable to poor segmentation in both systems. The perfect separation system has an EER of 11.8% compared to an EER of 15.4% for the external segmentation system. The

[4] The means and variances for this normalization are estimated from training data using utterances with the correct handset label and a duration of about 30 s.

perfect separation DET also highlights that even with the segmentation task removed, errors are not negligible, indicating there are substantial gains to be made on the core detection system.

Finally, the single-speaker detection performance using only the individual sides of the multispeaker conversations is shown in Fig. 6 as the lower dashed-dot line. The EER of the single-speaker detection system is 9.5% compared to 11.8% for the perfect separation curve. This increase in error can be attributed to the additional maximum function used in the perfect separation system to produce a single score for the entire multispeaker conversation. Considering that the two scores coming into the maximum function are outputs from two independent detectors each operating at $(P_{\text{miss}}, P_{\text{fa}})$, then it can be shown that the corresponding operating point after the maximum function is [5]

$$\widehat{P}_{\text{miss}} = P_{\text{miss}} * (1 - P_{\text{fa}})^{N-1} \quad \text{and} \quad \widehat{P}_{\text{fa}} = 1 - (1 - P_{\text{fa}})^{N}, \tag{9}$$

where $N = 2$. Applying this transform to the single-speaker DET curve produces an almost identical match to the perfect separation DET curve.

6. SPEAKER TRACKING EXPERIMENTS

In this section we compare the use of internal and external segmentation for speaker tracking. The performance of the speaker tracking systems is evaluated on the two-speaker, conversational, telephone speech from the 1999 NIST evaluation [1]. The test conversations for the tracking experiments are a subset of the conversations used in the detection experiments. In the tracking experiments 1000 of the 1723 test conversations are used and the number of imposter speakers is reduced from 20 per conversation to 2 per conversation. The external segmentation system has better tracking performance than the internal segmentation system, but it is shown that performance can still be substantially improved by better separating the two speakers.

Figure 7 shows the DET plots of tracking performance for our primary and secondary systems in the 1999 NIST evaluation. The primary system is a fixed-segment version of the internal segmentation system that uses a 2.5-s smoothing filter and reports scores every 0.25 s. The secondary system is the complete internal segmentation system as described in Section 3. For the fixed-segment system, segment sizes of 0.5–3.5 s were examined and about 2.5 was optimal on the development data. A longer segment gave better performance for false alarm rates greater than 2–3% but the cost function for the NIST task required operating at about a 1% false alarm rate. We tested various reporting intervals and determined that a 0.25-s reporting interval gave the best overall performance. This system performs slightly better than the internal segmentation system for false alarm rates less than 50%. The EER for this system is 26.7% while the EER for the internal segmentation system is 27.7%.

[5] These equations hold in general for a stack of N independent detectors.

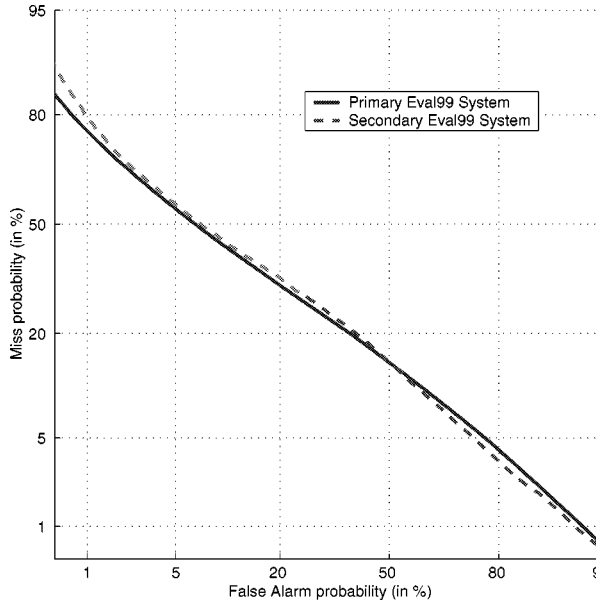

FIG. 7. Two-speaker tracking. The solid line is the fixed segmentation system and the dashed line is the internal segmentation system.

Neither of these systems uses handset normalization, as it was not found to help performance.

We do not show the 90% confidence intervals around the EERs for the tracking experiments. These experiments are scored by integrating the miss and false alarm regions over time so discrete, independent trials are not clearly defined for application of Eq. (8).

The external segmentation system for tracking uses the same procedure for clustering the data as does the external segmentation system for two-speaker detection. We tested various initial segment durations from 0.25 to 1.25 s to determine which was ideal for the tracking system and found that system performance varied only slightly as we varied this parameter. We also varied the number of clusters from two to six and found that this parameter did not have a large impact on system performance. We then chose a 0.5-s initial segment duration and three clusters since this appeared to have the best performance. We found that a small performance gain could be achieved by estimating the handset for each cluster and applying HNORM. This gave a 2–3% reduction of the miss rate for false alarms between 5 and 20% although it gave no improvement in other regions of the DET curve.

In Fig. 8 the DET curves for the above system are shown for two different methods of handling the silence regions. For the dashed curve the silence regions are given an arbitrarily low score and for the solid curve the silence regions are scored in the same manner as the other regions but the score is then clipped to 1.0. That is if the score is greater than 1.0 it is given the value 1.0. In the first case, when silence is given an arbitrarily low score, the miss rate has a floor of about 7%. This indicates that the silence marks generated by our speech

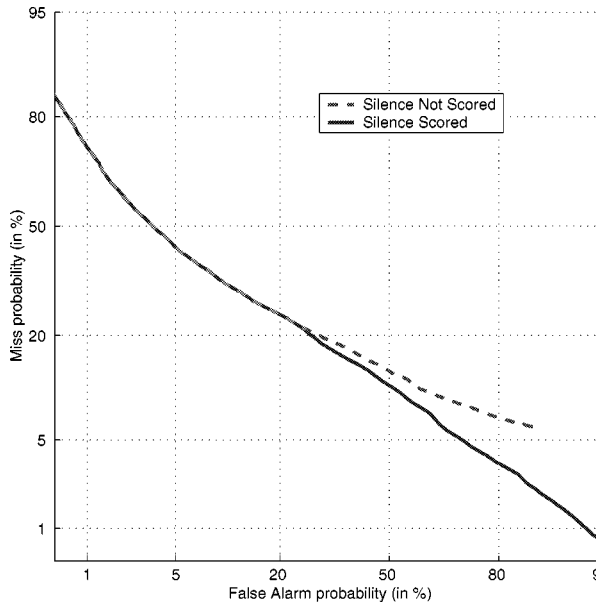

FIG. 8. Two-speaker tracking external segmentation. The dashed line is giving silence regions an arbitrarily low score. The solid line is scoring silence regions but clipping the score.

detector do not match the silence in the answer key. To account for this problem we score the silence region and clip the score to keep regions we have scored as silence from false alarming too readily. As seen by the solid curve this approach provides a lower miss rate for false alarm rates above 20%.

The performance of the fixed and external segmentation systems is compared along with the performance of ideal segmentation in Fig. 9. The thick dashed line is for the fixed segmentation system while the thick solid line is for the best external segmentation system which was more recently developed. This new system has a substantial improvement in performance for false alarm rates between 5 and 20%. The thin solid line is the performance of an ideal segmentation system. In the ideal segmentation there are four regions generated from the answer key containing: speaker A only, speaker B only, the overlap of speaker A and B, and silence. The first three regions are scored as when automatic external segmentation is used and the silence region is given an arbitrarily low score. The ideal segmentation system indicates the performance of the GMM-UBM scoring system without regard to the problem of segmenting the data. It shows that a great deal of performance improvement can be gained by improving segmentation of the multispeaker data.

7. DISCUSSION

There are two general observations we make from the experiments. First, it appears for both detection and tracking, the use of the external segmenter

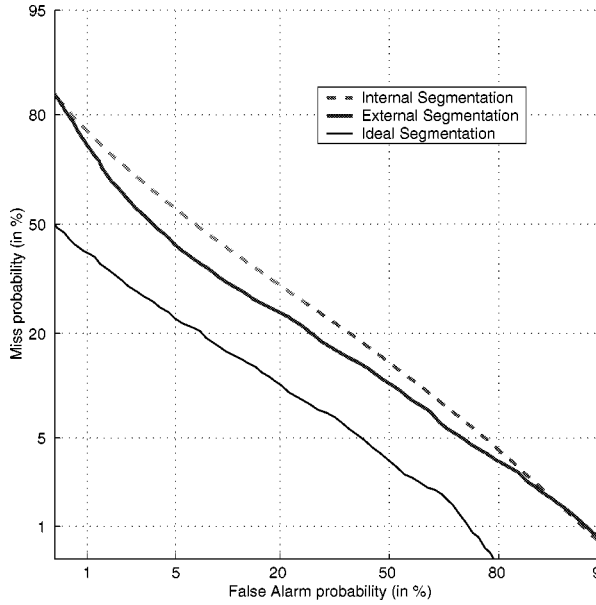

FIG. 9. Comparison of two-speaker tracking systems. The thin black line is tracking using the ideal segmentation. The dashed line is fixed segmentation and the thick solid line is the clustering based system.

gives better performance than the use of internal segmentation. This relative improvement is greater for detection than for tracking. It appears that in the internal segmentation system the use of the time-varying log-likelihood ratio function to both determine segments and validate those segments reduces performance. In the external segmentation systems, given reasonable performance from an external segmenter/clusterer, the log-likelihood ratio function is used only to validate precomputed segments.

Our second observation is that, as in the single-speaker detection task, the application of HNORM greatly improves performance for multispeaker detection. For tracking, HNORM provided none to minor improvement. One possible explanation for this difference is that in detection multiple segments throughout a file can be agglomerated together for computing a final detection score. In tracking local detection scores over short duration segments are required. It has been observed on single-speaker detection experiments that HNORM is not very effective for short duration (< 3 s) speech segments.

In further experiments using external segmenters, we also examined the use of gender identification and handset identification as methods of segmenting a two-speaker audio file. These methods, of course, can only be used for mixed gender and mixed handset conversations. The gender identification system actually performed better than the automatic clustering system on the mixed gender conversations from the 1999 evaluation data. Handset identification, on the other hand, was not a reliable method of segmenting the mixed handset conversations.

8. CONCLUSIONS

In this paper we have presented approaches to speaker detection and tracking with multispeaker audio. We have developed systems for both tasks using internal and external segmentation techniques and applied them to the 1999 NIST evaluation data. From the experiments, we found that the use of an external segmentation approach produces improved performance over an internal segmentation approach for both detection and tracking. While these systems produce state-of-the-art performance on the tasks, there is considerable room for improvement.

Two factors dominate the performance of both detection and tracking in multispeaker speech: the quality of the segmentation and the underlying likelihood ratio scoring. Comparison of results from our best performing systems to the ideal segmentation systems indicates that there does indeed exist room for improvement in the segmentation. The current external segmenter which uses blind clustering, is a simple approach which can be refined using perhaps multiple segmentation passes as is done in [12]. However, even with the segmentation component removed, performance is far from perfect indicating a real need for improvement in the underlying single-speaker detection scoring. With the introduction of HNORM to the multispeaker task we have improved robustness to handset variability, but our modeling and recognition are still vulnerable to nonspeaker variabilties. They rely on acoustic measurements made over short durations and these features are vulnerable to changes in the acoustic environment. Future work will concentrate on improving the robustness of the underlying adapted GMM-UBM system and also on the introduction of more complex features that are less vulnerable to changes in the acoustic environment, such as speaking rate and interactions between speakers.

REFERENCES

1. Martin, A. and Przybocki, M., The NIST 1999 Speaker Recognition Evaluation—An overview, *Digital Signal Process.* **10** (2000), 1–18.
2. *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
3. Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M., The DET curve in assessment of detection task performance. In *Proceedings of the European Conference on Speech Communication and Technology,* 1997, pp. 1895–1898.
4. Reynolds, D. A., Comparison of background normalization methods for text-independent speaker verification. In *Proceedings of the European Conference on Speech Communication and Technology,* September 1997, pp. 963–967.
5. Reynolds, D. A., Quatieri, T. F., and Dunn, R. B., Speaker verification using adapted Gaussian mixture models, *Digital Signal Process.* **10** (2000), 19–41.
6. Reynolds, D. A., HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* April 1997, pp. 1535–1538.
7. Soong, F. and Rosenberg, A., On the use of instantaneous and transitional spectral information in speaker recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1986, pp. 877–880.
8. Reynolds, D. A., *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification,* PhD thesis, Georgia Institute of Technology, September 1992.

9. Hermansky, H., Morgan, N., Bayya, A., and Kohn, P., RASTA-PLP speech analysis technique. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* March 1992, pp. I.121–I.124.
10. Gish, H., Schmidt, M., and Mielke, A., A robust, segmental method for text-independent speaker identification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1994, pp. I.45–I.148.
11. Rosenberg, A., Magrin-Chagnolleau, I., Parthasarathy, S., and Huang, Q., Speaker detection in broadcast speech databases. In *Proceedings of the International Conference on Spoken Language Processing,* 1998.
12. Wilcox, L., Chen, F., Kimber, D., and Balasubramanian, V., Segmentation of speech using speaker identification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1994, pp. I.161–I.164.
13. Gish, H., Siu, M. H., Rohlicek, R., Segregation of speakers for speech recognition and speaker identification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1991, pp. II.873–II.876.
14. Chen, S. and Gopalakrishnan, P., Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop,* 1998, http://www.nist.gov/speech/proc/darpa98/index.htm.
15. Jin, H., Kubala, F., and Schwartz, R., Automatic speaker clustering. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop,* 1998, http://www.nist.gov/speech/proc/darpa98/index.htm.

ROBERT DUNN received a B.S. in electrical and computer engineering (with highest honors) from Northeastern University in 1991 and he received a S.M. in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT) in 1995. He joined the Speech Systems Technology Group (now the Information Systems Technology Group) at MIT Lincoln Laboratory in 1991, where he is currently a member of the technical staff. In 1997 and 1998 he worked on the development of speech coding technology for Voxware, Inc. His research interests include speaker identification, low rate speech coding, and audio signal enhancement.

DOUGLAS REYNOLDS received the B.E.E. (with highest honors) in 1986 and the Ph.D. in electrical engineering in 1992, both from the Georgia Institute of Technology. He joined the Speech Systems Technology Group (now the Information Systems Technology Group) at the Massachusetts Institute of Technology Lincoln Laboratory in 1992. Currently, he is a senior member of the technical staff and his research interests include robust speaker identification and verification, language recognition, speech recognition, and general problems in signal classification. He is a senior member of the IEEE and a member of the IEEE Signal Processing Society Speech Technical Committee.

THOMAS F. QUATIERI received the B.S. (summa cum laude) from Tufts University, Medford, Massachusetts, in 1973, and the S.M., E.E., and Sc.D. from the Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, in 1975, 1977, and 1979, respectively. He is currently a senior member of the technical staff at MIT Lincoln Laboratory, Lexington, Massachusetts, involved in digital signal processing for speech and audio modification, coding, and enhancement and for speaker recognition. His interests also include nonlinear system modeling and estimation. He has contributed many publications to journals and conference proceedings, written several patents, and coauthored chapters in numerous edited books. He holds the position of lecturer at MIT, where he has developed the graduate course *Digital Speech Processing*. Dr. Quatieri is the recipient of the 1982 Paper Award of the IEEE Acoustics, Speech, and Signal Processing Society for the paper, "Implementation of 2-D Digital Filters by Iterative Methods." In 1990, he received the IEEE Signal Processing Society's Senior Award for the paper, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," and in 1994 won this same award for the paper, "Energy Separation in Signal Modulations with Application to Speech Analysis," which was also selected for the 1995 IEEE W.R.G. Baker Prize Award. He was a member of the IEEE Digital Signal Processing Technical Committee, he served on the steering committee for the biannual Digital Signal Processing Workshop from 1983 to 1992, and was Associate Editor for the *IEEE Transactions on Signal Processing* in the area of nonlinear systems. He is also a fellow of the IEEE and a member of Sigma Xi and the Acoustical Society of America.