Digital Signal Processing **10**, 55–74 (2000)
doi:10.1006/dspr.1999.0363, available online at http://www.idealibrary.com on IDEAL®

Data-Driven Temporal Filters and Alternatives to GMM in Speaker Verification

Narendranath Malayath*, Hynek Hermansky*,†, Sachin Kajarekar*, and B. Yegnanarayana‡

*Oregon Graduate Institute of Science and Technology, Portland, Oregon; †International Computer Science Institute, Berkeley, California; and ‡Indian Institute of Technology Madras, Chennai, India

E-mail: naren@ece.ogi.edu, hynek@ece.ogi.edu, sachin@ece.ogi.edu; yegna@iitm.ernet.in

Malayath, Narendranath, Hermansky, Hynek, Kajarekar, Sachin, and Yegnanarayana, B., Data-Driven Temporal Filters and Alternatives to GMM in Speaker Verification, *Digital Signal Processing* **10** (2000), 55–74.

This paper discusses the research directions pursued jointly at the Anthropic Signal Processing Group of the Oregon Graduate Institute and at the Speech and Vision Laboratory of the Indian Institute of Technology Madras. Current methods for speaker verification are based on modeling the speaker characteristics using Gaussian mixture models (GMM). The performance of these systems significantly degrades if the target speakers use a telephone handset that is different from that used while training. Conventional methods for channel normalization include utterance-based mean subtraction (MS) and RelAtive SpecTrAl (RASTA) filtering. In this paper we introduce a novel method for designing filters that are capable of normalizing the variability introduced by different telephone handsets. The design of the filter is based on the estimated second-order statistics of handset variability. This filter is applied on the logarithmic energy outputs of Mel spaced filter banks. We also demonstrate the effectiveness of the proposed channel normalizing filter in improving speaker verification performance in mismatched conditions. GMM-based systems often use thousands of mixture components and hence require a large number of parameters to characterize each target speaker. In order to address this issue we propose an alternative to GMM for modeling speaker characteristics. The alternative is based on speaker-specific mapping and it relies on a speaker-independent representation of speech.

Key Words: modulation spectrum; temporal processing; speaker verification; channel variability; data-driven filter design

1051-2004/00 $35.00

1. INTRODUCTION

Speech signal carries information from three main sources. These are (i) linguistic information source, (ii) speaker-specific information source, and (iii) source of information about the environment. In text-independent speaker verification we are interested in effectively modeling the speaker information contained in the speech signal. The other information sources are harmful for the task.

The task of speaker verification is to detect whether or not a given speech segment has been spoken by a claimed target speaker or not. Systems that use mixture of Gaussian functions for characterizing the distribution of acoustic vectors of target speakers have been shown to achieve good verification accuracy [14]. The individual components of the GMM represent different regions of the acoustic space. The unwanted linguistic variability of the acoustic features is suppressed by the simultaneous use of two GMMs, one is a speaker-specific GMM (λ_s), modeling the acoustic space of a given speaker, the second is the so-called universal background model (λ_b) [15]. The universal background model (UBM) is a speaker-independent model, trained using the speech data of a large number of speakers. Thus the UBM represents a speaker-independent distribution of the feature vectors. Speaker-specific models are obtained by maximum a posteriori (MAP) adaptation of the UBM with the data of target speakers. During the verification phase, the test utterance is scored using the UBM and the speaker-specific model corresponding to the target speaker. The claim is rejected or accepted by comparing the log likelihood ratio with a threshold θ as illustrated by

$$\sum_i \ln \frac{p(x_i|\lambda_s)}{p(x_i|\lambda_b)} \underset{\text{accept}}{\overset{\text{reject}}{\lessgtr}} \theta, \tag{1}$$

where x_i is a feature vector extracted from the ith frame.

The current approaches to speaker verification have some draw backs. In this paper we address two of them, namely,

– It has been observed that a reasonable verification accuracy can be achieved if the speaker uses the same handset and telephone line for training and testing [14]. On the other hand, if the speaker uses a different telephone handset while testing, the verification error increases by four to five times [4, 13].

– GMM-based systems use thousands of mixture components making it necessary to store a large number of parameters. Hence we identify the large size of the GMM models as the second issue to be addressed in this paper.

The focus of the first part of this paper is on feature processing methods for increasing the robustness of speaker verification systems in the presence of channel variability.

The variability introduced by microphones in particular and channels in general could significantly degrade the performance of both speech-recognition and

speaker-verification systems. In automatic speech recognition (ASR) systems it is typically possible to train the recognizers with speech recorded using different telephone handsets and this makes the system relatively insensitive to handset variability. On the other hand, in speaker verification, a statistical model must be designed from a relatively small amount of speech data to represent the acoustic features of the target speaker, and it is not always practical to have training utterances collected from multiple telephone handsets. This results in models which are highly biased toward the handset used to record the training utterance. Hence features that are robust to channel variability is of significance in speaker verification. In this paper we introduce a data-driven method for designing channel normalizing filters. *We call it data-driven because the method of design optimizes an objective function and the optimization is done over a significantly large amount of speech data*. The resultant filter is applied on the temporal trajectories of logarithmic spectral energies. Any stationary convolutive distortion will be an additive component in the logarithmic spectral energy domain, making it convenient for alleviating channel variability. Methods for processing temporal trajectories of logarithmic energy has already been proven to be effective in dealing with channel variability [8, 21].

The second part of this paper investigates an alternative approach to GMM-based speaker-verification systems. The proposed method is based on transforming a speaker-independent feature extracted from the signal to a speaker-dependent representation [6, 10]. The speaker-independent feature can be viewed as a parallel to the speaker-independent model used by the GMM-based systems. The mapping is implemented using a neural network trained for each of the target speakers. The speaker-specific mapping is shown to have a significantly smaller number of parameters as compared to GMM-based systems.

2. TEMPORAL PROCESSING FOR CHANNEL NORMALIZATION

2.1. Introduction to Temporal Processing

This section provides a brief introduction to temporal processing techniques used to extract features from speech. For a detailed review refer to [2, 7].

Temporal processing in the context of this paper means modification of temporal trajectory of a spectral component. This spectral component is the output of an auditory-like (Mel or Bark) filter. Let us denote the spectrogram resulting from short-time analysis as $S(\omega_k, t_i),\ k = 1, 2, \ldots, N,\ i = 1, 2, \ldots, T$, where N and T is the number of frequency bands and the number of time steps used for the short-time analysis, respectively. The temporal processing is done on logarithm of the squared magnitude, $S_l(\omega_k, t_i) = \log(|S(\omega_k, t_i)|^2)$. Then, $S_l(\omega_k, t_i)$, $i = 1, 2, \ldots, T$, is the time trajectory of the logarithmic energy corresponding to the kth frequency band. The power spectrum of such a time trajectory is referred to as the modulation spectrum [2]. The Nyquist frequency of modulation spectrum is given by $0.5/(t_i - t_{i-1})$. Typically, $t_i - t_{i-1}$ (which is the window shift in the short-time analysis) is 10 ms. Hence the Nyquist

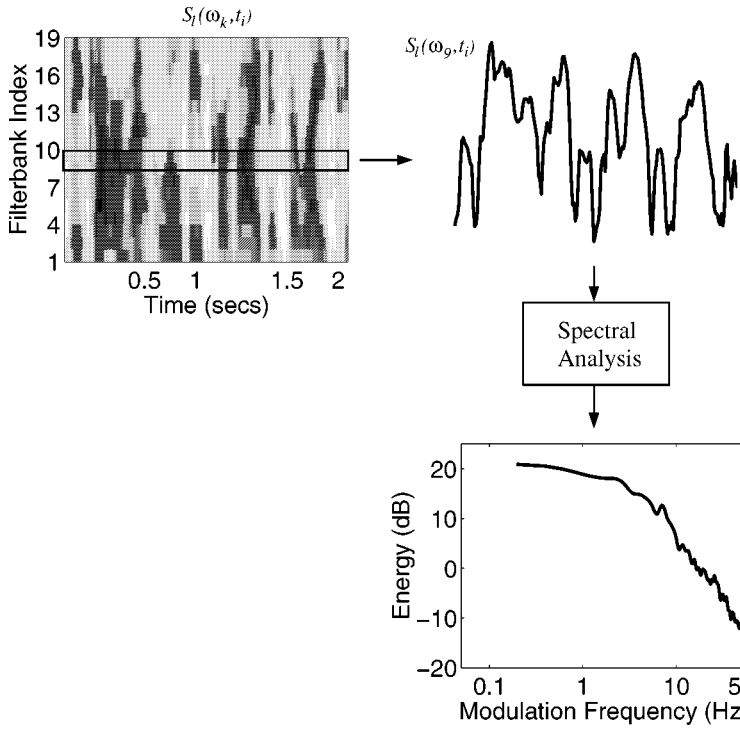

FIG. 1. Illustration of the notion of modulation spectrum of speech.

frequency of the modulation spectrum is 50 Hz. Figure 1 illustrates the concept of modulation spectrum.

It is straightforward to show that any convolutive distortion of the signal affects the mean of the time trajectory (direct current or the DC component of the modulation spectrum). Hence the mean subtraction (MS) operation applied on the log-magnitude spectrum (or any of its linearly transformed versions like cepstrum), popularly known as cepstral mean subtraction (CMS) can be viewed as temporal processing which removes the DC component of the modulation spectrum. MS has been shown to make the features robust to channel variability [1, 18]. More recently RASTA processing was introduced as an on-line alternative to MS [8]. The RASTA filter attenuates modulation frequency components below 1 Hz and above 10 Hz. Thus it not only eliminates stationary and slowly varying convolutive distortions but also eliminates fast varying (higher than 10 Hz) modulation frequency components. RASTA-like filters have enjoyed considerable success in dealing with channel mismatches in ASR [8]. Recently, it was shown that modulation frequency components below 1 Hz are important for speaker verification [21]. This suggested the need for an alternative channel normalizing filter for speaker verification. The studies on the effect of temporal filtering on speaker-verification performance has shown that simultaneous use of MS and a low-pass filter with 10 Hz cutoff yields improvement in verification performance when there are channel mismatches [21].

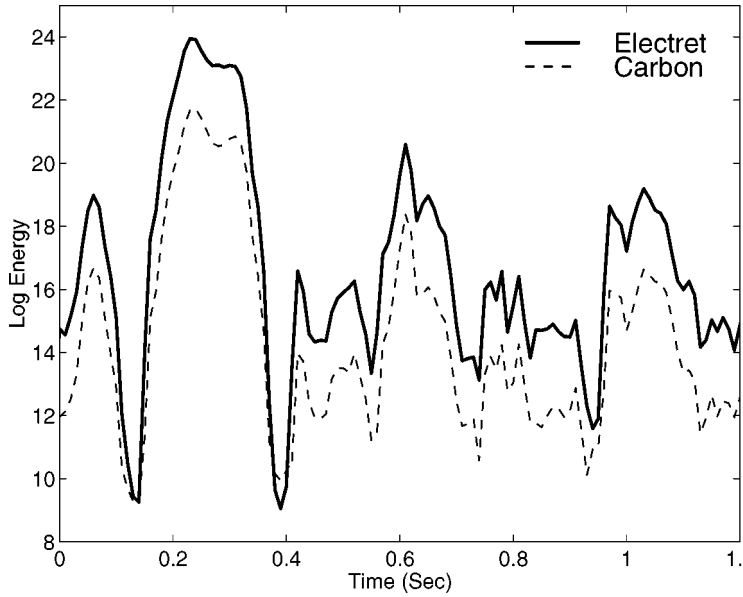

FIG. 2. Temporal trajectories of logarithmic energy from the 9th Mel frequency bank extracted from the same speech segment passed through carbon button and electret type microphones.

In the current work we suggest a data-driven method for automatic design of a temporal filter, which would suppress channel variability. This method avoids the use of computationally expensive and time-consuming speaker-verification experiments [21] in order to design appropriate temporal filters.

2.2. Characteristics of Channel Variability

Figure 2 illustrates some effects of channel mismatches on an utterance passed through two different microphones. The figure shows the temporal trajectory of logarithmic energy corresponding to the 9th Mel filter bank (1.1 kHz). One microphone is of carbon button type and the other is of electret type. These utterances are taken from the HTIMIT [16] data base, which consists of TIMIT sentences passed through 10 different microphones. The major difference between the two channels is the difference in the gain, which shows up as a DC shift in the logarithmic domain. Note that this gain difference will be nonuniform across frequency channels, depending on the frequency characteristics of the two microphones. It can also be noted that the gain difference is nonuniform across different segments of the utterance. Hence it is clear that the variation in microphone type affects the temporal trajectory of logarithmic energy in a complex manner. The following section develops a method to mathematically analyze these variations in order to design filters that suppress channel variability.

2.3. Channel Normalizing Filter Design

The objective is to design a filter which by acting on the time trajectories of logarithmic energies will minimize the variability introduced by different

microphones. The initial short-time processing using a 10 ms window shift yields logarithmic energies from 19 Mel spaced filter banks, which is represented by $S_l(\omega_k, t_i)$, $k = 1, 2, \ldots, 19$. Mean of these feature vectors were removed from each of the utterances to compensate for stationary convolutive distortions. A one second long time trajectory corresponding to the kth Mel filter is denoted by

$$X_{kt} = \left[S_l(\omega_k, t-50) \quad S_l(\omega_k, t-49) \quad \ldots \quad S_l(\omega_k, t+50) \right]^T,$$

and it represents the signal that needs to be filtered. Since the window shift used in the short-time analysis is 10 ms, X_{kt} is a 101-dimensional vector. The temporal filtering operation is represented by

$$Y_{kt} = X_{kt}^T h_k,$$

where h_k is the vector representing the time-reversed impulse response of the implied filter and Y_{kt} is the filtered signal. This filtering operation can be considered as projecting the vector X_{kt} onto the direction h_k.

For channel normalization, h_k should point to the direction in the feature space where the variability due to microphone mismatches is minimum. In order to prevent trivial solution ($h_k = 0$) we impose the additional constraint that h_k should retain as much of the relevant signal variability as possible while suppressing the channel variability. The design criteria can be achieved by selecting h_k to maximize signal-to-noise ratio, ρ_k, given by

$$\rho_k = \frac{h_k^T \Sigma_{sk} h_k}{h_k^T \Sigma_{nk} h_k}, \tag{2}$$

where Σ_{sk} is the covariance of the desired signal and Σ_{nk} is the covariance of the noise. Since channel variability is unwanted it is referred to as noise. It is fairly straightforward to show that the quantity ρ_k is maximized by setting h_k to e_k, which corresponds to the eigen vector with largest eigen value of the generalized eigen value problem

$$\Sigma_{sk} e_k = \lambda_k \Sigma_{nk} e_k. \tag{3}$$

This design process is repeated for each of the 19 Mel filter banks, thus yielding 19 temporal filters (e_k, $k = 1, 2, \ldots, 19$), one corresponding to each of the 19 bands. It is evident from the above discussion that for the proposed filter design method, we need only the second-order statistics of the channel and the desired signal variability that is required in order to design channel normalizing filters.

2.3.1. Estimation of channel variability. This section describes a method for estimating the second-order statistics of the variability introduced by telephone handsets. A subset of HTIMIT data base [16], which consists of TIMIT sentences passed through 10 different microphones and recorded synchronously, is used for this purpose. Our subset consisted of speech passed through four carbon button microphones and four electret microphones. These utterances were aligned using a correlation-based measure [21]. Temporal feature vectors extracted from speech recorded through ith and the jth

microphones are denoted by $X_{kt}(i)$ and $X_{kt}(j)$. It should be noted that $X_{kt}(i)$ and $X_{kt}(j)$ are extracted from the same sentence spoken by the same speaker in the same phonetic context. The only difference between $X_{kt}(i)$ and $X_{kt}(j)$ is that the speech signal from which they were extracted was recorded through two different telephone handsets. Hence the difference vector, d_k between these two temporal vectors should point to the direction of microphone variability. A set of these difference vectors,

$$d_k = X_{kt}(i) - X_{kt}(j), \qquad \forall i, j;\ i \neq j, \tag{4}$$

which point in the direction of channel variability is computed over the entire HTIMIT database. These difference vectors were extracted from various combinations of microphones, which included carbon–carbon (intracarbon), electret–electret (intraelectret), and carbon–electret. Hence this ensemble of difference vectors covers a variety of mismatches. This computation is independently done on all the 19 Mel frequency banks. The covariance of the channel variability,

$$\Sigma_{nk} = E\left[d_k d_k^T\right], \tag{5}$$

required for the design of channel normalizing filters is then estimated. About 2,000,000 difference vectors were used to estimate the above 101×101 channel covariance matrix.

2.3.2. Estimation of desired variability. For the design of a channel normalizing filter we consider the variability introduced by a channel as the undesired source of variability. As discussed in the beginning of this section (Eq. (3)) in order to yield a nontrivial solution the minimization of channel variability must be carried out while preserving as much of the desired signal variance as possible.

Since we are after text-independent speaker verification, it is tempting to consider the variability introduced by various speech sounds (which we call phonetic variability) as undesirable. However, the simultaneous use of UBM and the speaker-specific GMM effectively alleviates phonetic variability [15]. Moreover, since each of the mixture components are independently adapted using speaker-specific data, the GMM-based speaker verification system can potentially capture phoneme specific speaker characteristics. Hence, removing phoneme variability could potentially degrade the performance. Therefore, in our filter design we are considering the variability introduced by phonetic classes as the signal variance which needs to be preserved.[1] For the filter to be optimal for speaker verification we must constrain the solution of Eq. (3) in order to prevent the removal of speaker variability. This can be done by considering the combined speaker and phoneme variability as the desired source of variability. Even though the current design is not taking this aspect into account (hence in this respect is suboptimal), the capability of the filter in

[1] Even though the acoustic units defined by the UBM may not have an exact one to one correspondence with phonemes, we assume that preserving phonetic variability would lead to an improvement in the capability of the UBM in efficiently segmenting the acoustic space.

making speaker verification robust to channel mismatches is experimentally demonstrated in the following section.

Phonetic variability which needs to be preserved in the 101-dimensional temporal vector $X_{kt}(i)$ is estimated as follows. Each of these temporal vectors is labeled by the phoneme which is aligned to the center element. The class conditional mean of the phoneme, p, is given by

$$\mu_{pk} = \frac{1}{N_p} \sum_{\substack{t \in p \\ 1 \le i < 8}} X_{kt}(i),$$

where the index i represents the handset index which ranges from 1 to 8 as we are using TIMIT sentences recorded using 8 different telephone handsets. The phonetic variability corresponding to the kth channel is computed as the across class covariance Σ_{sk} given by

$$\Sigma_{sk} = \sum_{p} \frac{N_p}{N} [\mu_p - \mu][\mu_p - \mu]^T,$$

where N_p is the number of temporal vectors which are labeled as phoneme p, N is the total number of temporal vectors involved in the computation, and μ is the global mean of the temporal vectors. The channel normalizing filter can now be estimated for each of the frequency bands using the Eq. (3). The next section discusses the characteristics of the resulting filter.

2.3.3. Filter characteristics. Irrespective of the frequency band, these filters exhibit similar characteristics. This indicates that the relative characteristics of phoneme variability and channel variability are independent of the frequency band. This observation is supported by earlier work in designing temporal filters [2]. Hence for further discussion we will be describing only one of these filters. Frequency response of the filter extracted for the 9th Mel frequency band is shown in Fig. 3. Note that the since the filter was designed using mean normalized utterances it does not significantly attenuate DC. [2] The filter enhances the frequency components between 3 and 4 Hz. Spectral components above 5 Hz are significantly attenuated. This is one of the main differences between this filter and the conventional RASTA filter which has a pass band from 1 to 10 Hz.

For an objective measure of the effectiveness of the proposed filter, we computed the ratio of phoneme variance to channel variance before and after filtering. We call this measure signal-to-noise ratio (SNR). The SNR, ρ_k, corresponding to the kth frequency band after applying the filter f is defined by

$$\rho_k = 10 \log_{10} \left[\frac{f^T \Sigma_{sk} f}{f^T \Sigma_{nk} f} \right], \tag{6}$$

where Σ_{nk} is the channel covariance and Σ_{sk} is the phoneme covariance extracted from the kth frequency band. Figure 4 shows SNR as a function of

[2] NIST evaluation procedure allows for the off-line processing which is necessary for removing mean.

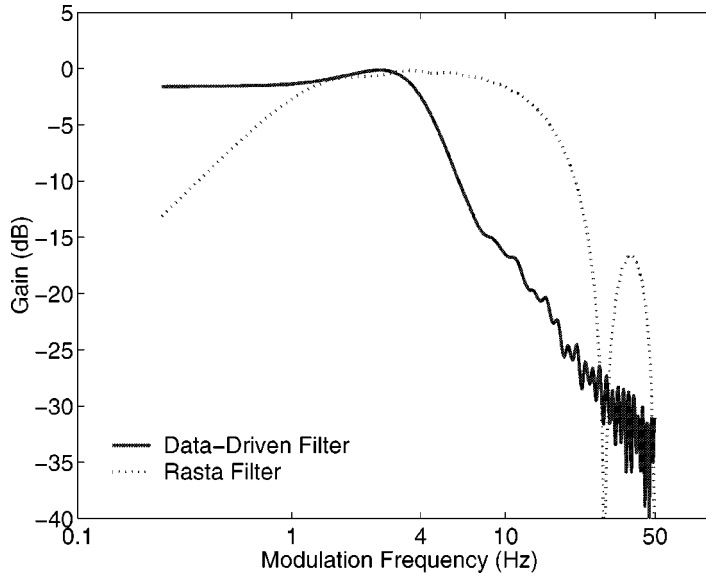

FIG. 3. Frequency response of the data-driven channel normalizing filter obtained from the 9th Mel frequency band. The figure also shows the frequency response of the classical RASTA filter.

19 Mel frequency bands before and after temporal processing. The RASTA filter significantly improves the SNR, and the proposed data-driven filter improves

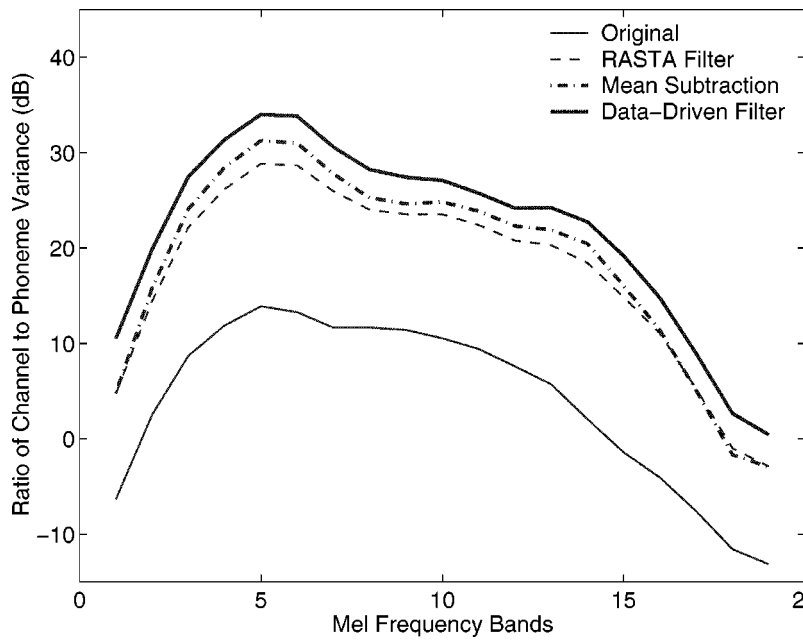

FIG. 4. Ratio of phoneme to channel variance as a function of the 19 Mel spaced filter-banks for various types of channel normalization.

the SNR even further. This illustrates the effectiveness of the proposed method in designing channel-normalizing filters.

2.4. Effect of Filtering on Text-Independent Speaker Verification

This section does a detailed analysis of the effect of the channel normalizing filters on text-independent speaker verification performance.

2.4.1. Description of the task and data base. Continuous telephone quality speech from the Switchboard-2 phase 3 corpus was used for all the speaker verification experiments described in this section. This data set was used by National Institute of Standards and Technology (NIST) in its 1999 official speaker verification evaluation [11]. The training data for each speaker consists of about 2 min of data collected from two separate (about 1 min long) sessions. Testing consists of utterances of durations anywhere between 1 s to 1 min.

The errors made by a speaker verification system can be of two types, false acceptance (verifying an imposter as the claimed target) and false rejection (rejecting a target speaker as an imposter). The trade-off between false acceptance rate (FAR or P_{FA}) and false rejection rate (FRR or P_{FR}) is determined by the threshold θ. A plot of FRR as a function of FAR is called the detection (DET) curve. Equal error rate (EER) which is defined as the FAR (or FRR) when the FAR is equal to the FRR, is used to evaluate the performance of the verification system. NIST evaluates the performance of the systems using a measure called decision cost function (DCF) where the cost of false alarms (C_{FR}) are 10 times more than the cost of false rejections (C_{FA}). The DCF is defined by

$$\mathrm{DCF} = C_{\mathrm{FR}} P_{\mathrm{FR}} P_{\mathrm{Target}} + C_{\mathrm{FA}} P_{\mathrm{FA}} P_{\mathrm{NonTarget}},$$

where $C_{\mathrm{FR}} = 10.0$, $C_{\mathrm{FA}} = 1.0$, $P_{\mathrm{Target}} = 0.01$ and $P_{\mathrm{NonTarget}} = 0.99$. For more details refer to [11]. Results were analyzed using both EER and DCF for the following two different test conditions.

– **Matched condition:** The training and testing utterances are collected from the same telephone handset. This condition is met only for the utterances of the genuine speakers (target trials). No such restrictions are placed on imposter trials.

– **Mismatched condition:** The test data is recorded through a handset which is of a different type compared to the one used for recording the training utterances. For example if the training utterance is recorded using a carbon button microphone then the test utterance of the target speaker is recorded using an electret type microphone. In mismatched condition also no restrictions are placed on imposter trials.

2.4.2. Feature extraction and modeling. Spectral energies from Mel spaced filter banks were derived by the processing used in Mel-cepstral analysis of speech [12]. For the speaker verification experiments 19 filter banks falling within the telephone bandwidth were used. The proposed data-driven temporal filter is then applied on each of the 19 trajectories of logarithmic energies

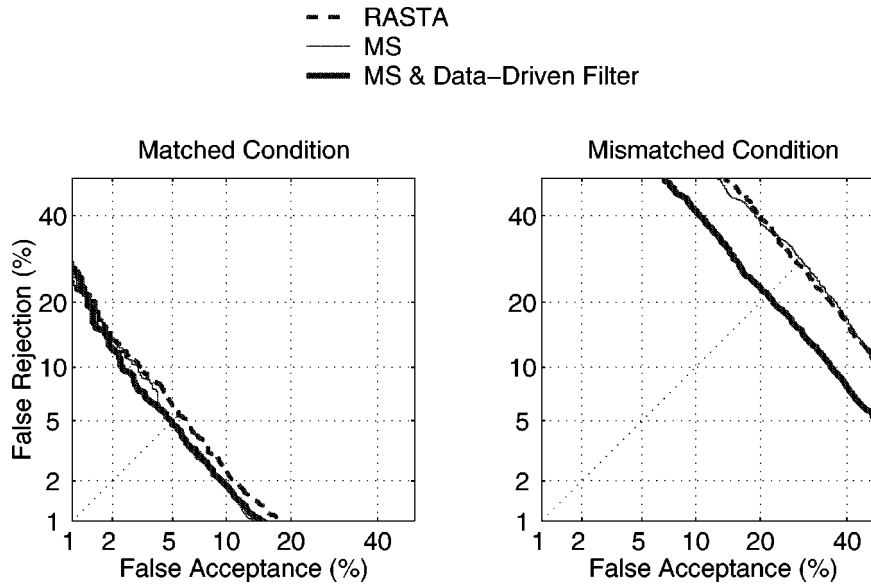

FIG. 5. Plot illustrating the significant reduction in error due to the data-driven temporal filter compared to the conventional mean subtraction (MS) and RASTA filtering.

after an utterance-based mean removal. From each of the 19 filtered time trajectories, delta features are computed [5]. The delta features are appended to the original feature vector to obtain a 38-dimensional feature vector. The distribution of feature vectors is then whitened using a Karhunen–Loeve (KL) transform computed from an independent data set. The first 36 components of the KL transform are retained for further processing. We have observed that whitening helps in reducing the number of mixture components without any degradation in performance. The baseline system uses only an utterance-based mean removal with appended delta and double delta features. The proposed filter is also compared with the standard RASTA filter.

For modeling the extracted feature we used the GMM–UBM paradigm which is described in [17]. A background model having 256 mixture components was trained using data from 80 speakers. This model was then adapted using the target speaker's data to obtain speaker-dependent models. For the implementation details of the system see [22].

2.4.3. Experimental results. Figure 5 illustrates the improvement due the data-driven temporal filter over the systems which use mean subtraction and RASTA filtering. In the mismatched condition, RASTA filtering performs as good as MS. The data-driven filter, which is applied on top of utterance-based mean subtraction, brings a significant reduction in error when there is a handset type mismatch. This clearly illustrates the effectiveness of the proposed filter in suppressing the handset variability. In matched condition all three techniques perform equally well. Table 1 compares the EER and minimum DCF resulting from the use of MS, RASTA, and the data-driven filtering in combination with MS. In mismatched condition, the data-driven filter reduces

TABLE 1

The Performance of Systems Using Mean Subtraction, RASTA Filtering, and the Data-Driven Filter.

Processing	Mismatched		Matched	
	EER	MDCF	EER	MDCF
MS	28.8%	0.084	4.9%	0.022
RASTA	27.9%	0.086	5.4%	0.027
MS and Data-Driven Filter	21.4%	0.078	4.9%	0.024

Note. The table provides the comparison using both equal error rate (EER) and minimum decision cost function (MDCF).

the equal error rate by more than 25%. The MDCF is also reduced by about 10% compared to RASTA. In matched condition the data-driven filter performs as well as MS, but RASTA causes a degradation in performance. This degradation is attributed to the removal of modulation frequency components between 0 and 1 Hz which have been shown to contain important information relevant for speaker verification [21].

Since the data-driven filter was designed using mean normalized feature vectors the filter does not attenuate DC component of the modulation spectrum and hence makes it necessary to be used in conjunction with mean subtraction. In many applications mean removal presents practical difficulties. For example, in order to estimate the mean it is necessary to wait until the end of the utterance. This might result in long delays which may not be acceptable for many applications. If the signal contains speech from multiple speakers recorded us-

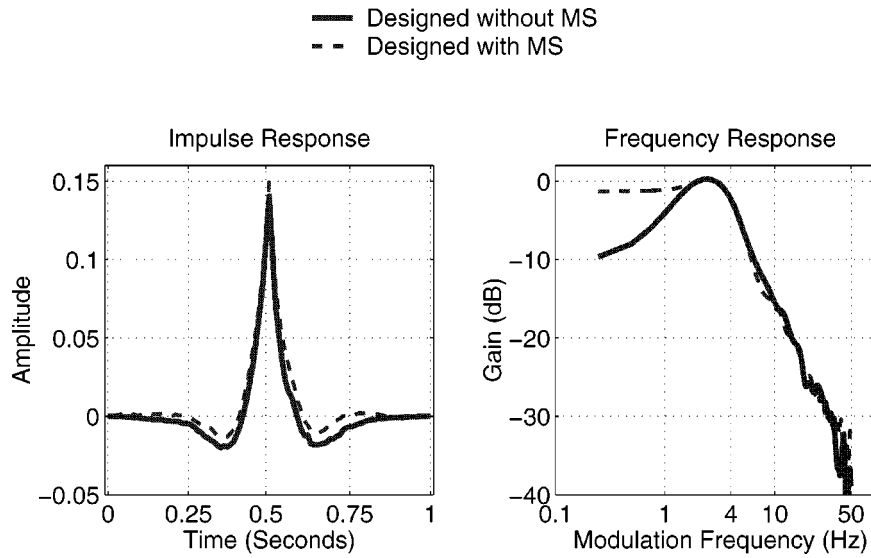

FIG. 6. Effect of mean removal on the frequency response of the data-driven filter.

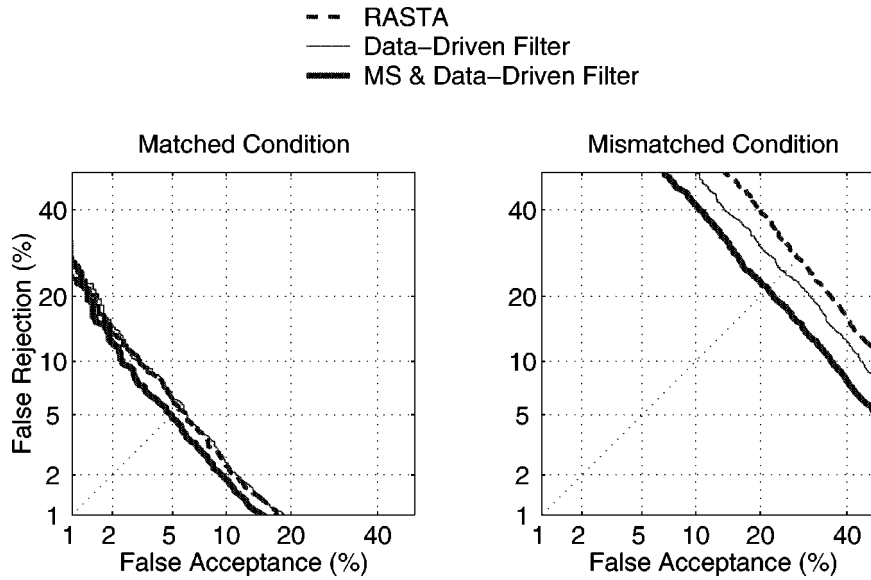

FIG. 7. Plot illustrating the effect of mean subtraction (MS) used in conjunction with the data-driven filter.

ing different handsets then the estimated mean will contain information about both the microphones and thus cannot be used for normalizing the channel variability. This motivates the need for designing a filter that will attenuate the DC component of the modulation spectrum and thus avoids the use of mean subtraction. For designing such a filter the procedure described in Section 2.3 was repeated using temporal trajectories from which means were not removed. Figure 6 compares this filter to the filter derived with mean removal. The only significant difference between the two filters is in the frequency characteristics below 1 Hz. The filter designed without mean subtraction attenuates modulation frequency components below 1 Hz. The performance of these two filters are compared with the RASTA filter in Fig. 7. From the DET curve it is clear that the data-driven channel normalizing filter performs better than the RASTA filter. But its performance is worse than that obtained by the simultaneous use of the channel normalizing filter and MS. This trend can be explained as follows. The channel normalizing filter used in conjunction with MS does not attenuate frequency components below 1 Hz (broken line in Fig. 6). The simultaneous use of this filter and MS would imply a frequency response which is dependent on the length of the utterance. The longer the utterance, the less the attenuation of components between 0 and 1 Hz. On the other hand, the channel normalizing filter that is used without MS, attenuates the DC component (solid line in Fig. 6). The frequency resolution of this filter is restricted to 1 Hz (since the number of taps are 101). This causes the filter to attenuate frequency components between 0 and 1 Hz irrespective of the length of the utterance. It has already been shown that modulation frequency components below 1 Hz are useful

for speaker verification [21], and hence their attenuation causes the degradation in speaker verification performance.

3. SPEAKER VERIFICATION BY SPEAKER-SPECIFIC MAPPINGS

3.1. Background

As mentioned in Section 1, GMM-based speaker verification systems use a large number of parameters to model speaker characteristics. In this section, we propose an alternative modeling method which would require a significantly smaller number of parameters, and it would also avoid the need for training a background model. Speech is produced by a constrained physical system. Hence, we believe that the variability introduced by different speakers is systematic [20] and can be modeled using fewer parameters than the current GMM-based systems.

3.2. The Mapping Approach

The proposed method aims at suppressing the linguistic information from the signal by employing two simultaneous representations of speech, one containing lesser speaker-specific information than the other. By exploiting the difference in the two speech representations, we attempt to focus on the speaker-specific information source. This is done through the following three steps.

1. Extract features vectors, **I**, from the speech of the target speaker which primarily contain linguistic information. This representation must be relatively speaker-independent. We call this set of features speaker independent (SI) representation.
2. Extract feature vectors, **D**, which carry both linguistic and speaker information. Let us call this set of features speaker-dependent (SD) representation.
3. Estimate a functional mapping, **M** between **D** and **I** such that $\langle(\mathbf{D} - \mathbf{M}(\mathbf{I}))^2\rangle$ is minimized.

Note that the mapping function attempts to transform the speaker-independent feature vector **I** into a speaker-dependent feature vectors **D**. Thus, if both the **I** and **D** carry the identical linguistic information, the mapping **M** should carry the information which is present in **D** and not in **I**, i.e., the speaker-specific information.

During the verification phase, these speaker-specific models (**M**) are used to accept or reject the identity claims of speakers. This is done by the following steps.

1. Derive the SI and the SD feature vectors from the test utterance.
2. Transform **I** using the claimed speaker's model to derive the estimate of the SD feature vector, $\hat{\mathbf{D}} = \mathbf{M}(\mathbf{I})$.
3. Compute a distance measure, **S**, between **D** and $\hat{\mathbf{D}}$.

Figure 8 illustrates the steps involved in the verification phase. If the speaker is genuine then there should be a good match between $\hat{\mathbf{D}}$ and **D**. This will be

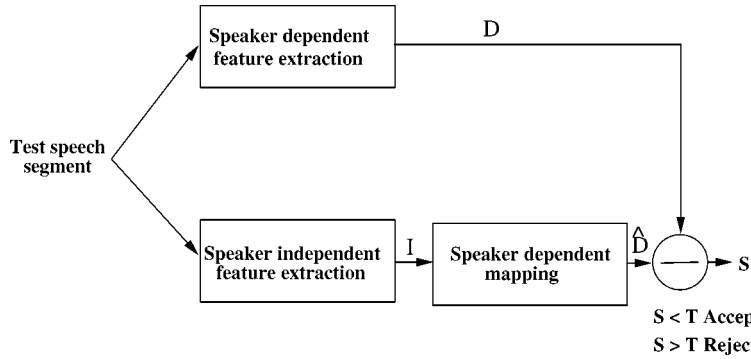

FIG. 8. Illustration of speaker verification by speaker specific mapping.

reflected in the distribution of **S** computed over the test segment available for verification. Thus, for example, the average of **S** should be small over a speech segment if the speech belongs to the genuine speaker. In the current system we accept or reject the claim based on the means of distributions, i.e., depending on whether the mean of **S** computed over the text segment falls below or above a predetermined threshold.

3.3. Speaker-Dependent Representation

Differences in vocal tract length is one of the major factors that introduce speaker characteristics in the speech signal. These differences affect the formant locations. This variability is thus in the details of the spectral envelope. It has been shown that a smoothed critical-band spectrum (smoothed by using a 5th order all-pole model) suppresses speaker variability [6]. Hence, higher order PLP features [6], LPC features, or Mel frequency warped filter-bank energies are all good candidates for the speaker-dependent representation.

3.4. Speaker-Independent Representation

In the proposed mapping approach, the text independence is achieved by the simultaneous use of SD and SI representations. The effectiveness of SI representation is critical to the performance of the verification system. In our initial experiments we have used lower order PLP features which are effective in suppressing the speaker information while preserving the important linguistic information [6]. Recently, we have initiated research in novel data-driven techniques for deriving speaker-independent features. One such technique is based on the use of Oriented Principal Component Analysis (OPCA) to estimate directions in the feature space which suppresses the speaker information [10]. This method first derives two vector spaces from a conventional feature space, one carrying primarily speaker information and the other carrying primarily linguistic information. These vector spaces are further used by OPCA to estimate a subspace, which optimally suppresses speaker information while preserving the linguistic information. The original features are then projected into this subspace yielding feature vectors, which are relatively speaker independent.

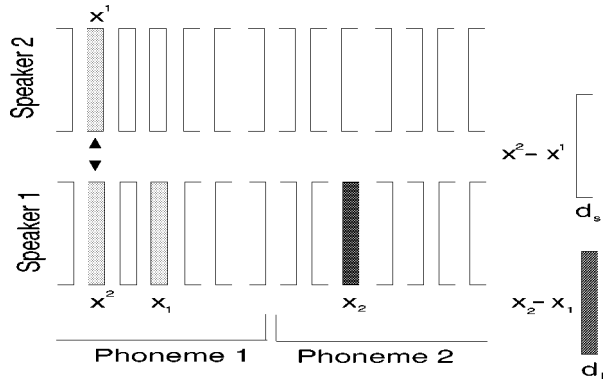

FIG. 9. Derivation of the difference vectors.

3.4.1. Decomposition of the feature space. In this section a method to decompose a conventional feature space (defined by PLP-cepstrum) into subspaces carrying mainly linguistic information and speaker information is presented. The initial feature representation $\mathbf{x}$ is the PLP-cepstrum and is considered as a random vector. Figure 9 shows the feature vectors extracted from a segment of speech from two speakers. The rectangular boxes represent feature vectors extracted from a frame of speech data. It is assumed that the segments of speech uttered by the two speakers are linguistically identical (same phonemes) and are perfectly time allingned. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the cepstral vectors from two different phonemes uttered by the same speaker. The difference vector carrying linguistic information is given by

$$\mathbf{d}_\mathrm{l} = \mathbf{x}_2 - \mathbf{x}_1. \tag{7}$$

By taking the difference between $\mathbf{x}_2$ and $\mathbf{x}_1$ the information which is common to $\mathbf{x}_2$ and $\mathbf{x}_1$ is removed. Hence the static (stationary) speaker characteristics and the channel effects are suppressed. Thus it can be concluded that the difference vector $\mathbf{d}_\mathrm{l}$ mainly carry information about the linguistic variability. Now consider the case where $\mathbf{x}^1$ and $\mathbf{x}^2$ represent the PLP-cepstrum extracted from the same phoneme uttered by two different speakers. Since $\mathbf{x}^1$ and $\mathbf{x}^2$ are features extracted from the speech signal corresponding to the same phoneme their difference will mainly contain speaker information. The difference vector representing speaker information is given by

$$\mathbf{d}_\mathrm{s} = \mathbf{x}^2 - \mathbf{x}^1. \tag{8}$$

Since $\mathbf{x}^1$ and $\mathbf{x}^2$ carry the same linguistic information the difference vector $\mathbf{d}_\mathrm{s}$ will mainly carry information about the speaker variability and the difference in the channel and environmental condition captured by the utterances of the two speakers. If the channel and environmental conditions captured by the speech signals of both the speakers are identical, then the random vectors $\mathbf{d}_\mathrm{l}$ and $\mathbf{d}_\mathrm{s}$ capture the linguistic and speaker variability, respectively.

3.4.2. Subspace-based feature extraction. The difference vectors $\mathbf{d}_l$ and $\mathbf{d}_s$ were extracted from the NTIMIT database using sets of sentences which were spoken by a set of speakers. The utterances were time aligned using DTW to compute the difference vectors corresponding to speaker variability, $\mathbf{d}_s$. The covariances of the difference vectors $\mathbf{d}_l$ and $\mathbf{d}_s$ are represented by $\mathbf{R}_l$ and $\mathbf{R}_s$, respectively. Since the objective is to maximize the variance caused by linguistic information and minimize the variance caused by speaker information, the objective function that we are interested in maximizing can be written as

$$\frac{E(\mathbf{d}_l^T \mathbf{e}_i)^2}{E(\mathbf{d}_s^T \mathbf{e}_i)^2} = \frac{\mathbf{e}_i^T \mathbf{R}_l \mathbf{e}_i}{\mathbf{e}_i{}^T \mathbf{R}_s \mathbf{e}_i} = \rho_i . \tag{9}$$

We are interested in finding the direction $\mathbf{e}_i$ which maximizes the signal-to-noise ratio, ρ_i. Deriving such directions (or projections) is nothing but the solution to the following generalized eigen value problem,

$$\mathbf{R}_l \mathbf{e}_{o_i} = \lambda_{o_i} \mathbf{R}_s \mathbf{e}_{o_i} . \tag{10}$$

This generalized eigen decomposition method is also known as oriented principal component analysis (OPCA) [3]. Now the original feature vectors $\mathbf{x}$ can be projected onto these eigen vectors to obtain the speaker-independent representation $\mathbf{I} = \mathbf{E}_o^T \mathbf{x}$, where $\mathbf{E}_o$ is a matrix whose columns are composed of the basis vectors. Note that since $\mathbf{I}$ is contained in a subspace of $\mathbf{D}$, $\mathbf{D}$ contains all the information that is captured by $\mathbf{I}$. Hence if $\mathbf{I}$ is truly speaker independent then the mapping $\mathbf{M}(\mathbf{I})$ will capture speaker information and will be independent of linguistic information. Since this method does not put any constraint on variability introduced by environment (like handset) we hope that both $\mathbf{I}$ and $\mathbf{D}$ will be affected by environment in a similar manner and hence the mapping $\mathbf{M}(\mathbf{I})$ will be independent of the acoustic environment.

3.5. Some Potential Advantages of Speaker-Specific Mapping over GMM

One of the conceptual advantages of the mapping-based approach is:

– Unlike the current mainstream speaker-verification techniques, which rely exclusively and rather blindly on large training databases, the mapping technique could in principle capitalize on the hard-wired knowledge in estimation of speaker-independent features from the speech signal. Thus if the representations $\mathbf{I}$ and $\mathbf{D}$ carry the same linguistic and environmental information then the mapping will capture only the speaker specific information. We are currently working toward the development of such features.

Some practical advantages are:

– The mapping-based method does not require the construction of a background model, avoiding the need for a large database of background-speakers.

– GMMs used for speaker verification have tens of thousands of free (trainable) parameters. The mapping-based method will typically have only a few hundred free parameters, yielding computational advantage.

TABLE 2

Equal Error Rates for Three Verification Experiments: Exp 1, Mapping from 5th to 12th Order PLP Cepstral Coefficients; Exp 2, Mapping from OPCA Features to 12th Order PLP Cepstral Coefficients; Exp 3, Baseline System Using a 128-Component Mixture Model

Testing condition		Exp 1	Exp 2	Exp 3
3 s	matched	23.1%	19.4%	11.3%
	mismatched	36.4%	30.7%	26.6%
10 s	matched	16.5%	12.7%	6.6%
	mismatched	32.7%	25.0%	18.6%
30 s	matched	12.5%	9.3%	4.9%
	mismatched	28.5%	22.5%	17.0%

3.6. Experimental Results

Speaker verification experiments were conducted on the SWITCHBOARD database with 40 target speakers which form a subset of the data used by NIST in 1998 speaker recognition evaluation [13]. Two minutes of speech was used for training the mapping models and segments of approximately 3, 10, and 30 s duration were used independently for verification. Results of these experiments for two different experimental setups are presented in Table 2. In the first experiment we used 7th order PLP cepstral coefficients as the SI representation and 14th order PLP cepstral coefficients were used as the SD representation. For the second experiment we used the oriented PCA features (using the projection into the subspace spanned by the first six oriented principal components [10]) as the SI representation. Since the multilayer feed-forward networks are universal approximators [9], we use them to estimate the mapping function. For both the experiments a neural network with a single hidden layer with 30 units was used for capturing the mapping. We notice that the OPCA features consistently outperform the conventional PLP features over all testing conditions. A GMM system was used as the baseline and the equal error rate is about half that of the mapping system in matched condition. The number of parameters used by the GMM system is 4992 ($256 \times 19 + 256 \times 19 + 256$), whereas the mapping system uses only 600 parameters ($6 \times 30 + 30 \times 14$).

3.7. Discussion

The results indicate that the performance of the mapping-based method is encouraging even though the error is still higher as compared to our GMM system. However, any complementary information given by the mapping system can be used to further improve the performance of the GMM system as described in [19]. From the table it is also evident that, compared to PLP features, the use of oriented PCA-based features as the speaker-independent representation, results in a significant reduction in error. This suggests that, by using better speaker-independent representation the performance of the proposed mapping-based speaker verification system can be improved.

4. SUMMARY

In this paper we have first proposed a method for normalizing channel mismatches for speaker-verification applications. The channel normalization is achieved through filtering the time trajectories of logarithmic filter-bank energies. A novel method to estimate the statistics of the variability introduced by the channel is also presented. The channel normalizing filter is designed to optimally suppress the channel variability under the constraint of preserving phonetic variability. The data-driven method uses only the second-order statistics and thus the solution is obtained by solving a straightforward eigen value problem. The filter emphasizes the modulation frequency components between 3 and 4 Hz. The gain of the filter drops off fairly sharply beyond 5 Hz. The proposed filtering method performs significantly better than the earlier methods including spectral mean subtraction and RASTA filtering. On the 1999 NIST speaker-verification task we observed about 25% relative error reduction in mismatched condition without any degradation in matched condition [11].

We have also proposed and investigated an alternative to the Gaussian mixture model (GMM) in text-independent speaker verification. This alternative is based on speaker-specific mapping. Compared to the conventional GMM system, the proposed mapping method uses a much smaller number of parameters to model speaker information. The performance of the new technique is still not on par with the GMM-based systems. However, there seems to be potential for improvement of the technique as our ability of deriving the speaker-independent information of speech improves. This has been demonstrated by speaker-independent speech representation based on oriented principal component analysis (OPCA), which has shown an advantage over the more conventional low-order PLP in speaker verification.

ACKNOWLEDGMENTS

This work was supported by DoD (MDA904-98-1-0521 and MDA904-99-1-0044). The authors thank the reviewers for their comments.

REFERENCES

1. Atal, B. S., Automatic recognition of speakers from their voices, *Proc. IEEE* **64** (1974), 460–475.
2. Avendano, C. and Hermansky, H., On the properties of temporal processing for speech in adverse environments. In *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics.* Mohonk Mountain House, New York, 1997, pp. 1–12.
3. Diamantaras, K. I. and Kung, S. Y., *Principal Component Neural Networks—Theory and Applications,* first ed. Wiley, New York, 1996.
4. Doddington, G. R., Speaker recognition evaluation—an overview and perspective. In *Proc. of Speaker Recognition and its Commercial and Forensic Applications, France,* 1998, pp. 60–66.
5. Furui, S., Speaker-independent isolated word recognition based on emphasized spectral dynamics. In *Proc. ICASSP, Tokyo, Japan,* 1986, pp. 1991–1994.
6. Hermansky, H., Perceptual linear predictive (PLP) analysis of speech, *JASA* **87**(4) (1990), 1738–1752.

7. Hermansky, H., Should recognizers have ears? *Speech Commun.* **25** (1998), 3–27.
8. Hermansky, H., and Morgan, N., RASTA processing of speech, *IEEE Trans. Speech Audio Process.* **2**(4) (1995), 578–589.
9. Hornik, K., Stinchcombe, M., and White, H., Multilayer networks are universal approximators, *Neural Networks* **2** (1989), 359–378.
10. Malayath, N., Hermansky, H., and Kain, A., Towards decomposing the sources of variability in speech. In *Proc. EUROSPEECH-97, Greece,* 1997, pp. 497–500.
11. Martin, A. and Przybocki, M., The NIST 1999 speaker recognition evaluation—An overview, *Digital Signal Processing* **10**(1) (2000), 1–18.
12. Mermelstein, P., Distance measures for speech recognition, psychological and instrumental. In *Pattern Recognition and Artificial Intelligence* (Chen, R. C. H., Ed.). Academic Press, New York, 1976.
13. Przybocki, M. A. and Martin, A. F., NIST speaker recognition evaluation—1997. In *Proc. of Speaker Recognition and its Commercial and Forensic Applications, Avignon, France,* 1998, pp. 120–123.
14. Reynolds, D. A., Speaker identification and verification using gaussian mixture models, *Speech Commun.* **17** (1995), 91–108.
15. Reynolds, D. A., Comparison of background normalization methods for text-independent speaker verification, In *Proc. EUROSPEECH-97, Greece,* 1997, pp. 963–966.
16. Reynolds, D. A., HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects. In *Proc. ICASSP, Munich,* 1997, pp. 1535–1538.
17. Reynolds, D. A., Quatieri, T. F., and Dunn, R. B., Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing* **10**(1) (2000), 19–41.
18. Schwarz, R., Anastasakos, T., Kubala, F., Makhoul, J., Nguyen, L., and Zavaliagkos, G., Comparitive experiments on large vocabulary speech recognition. In *Proc. of ARPA Workshop on Human Language Technology, Plainsboro, NJ,* 1993, pp. 1–12.
19. Sharma, S., Vermeulen, P., and Hermansky, H., Combining information from multiple classifiers for speaker verification. In *Proc. of Speaker Recognition and its Commercial and Forensic Applications, France,* 1998, pp. 115–119.
20. Umesh, S., Cohen, L., and Nelson, D., Frequency-warping and speaker-normalization. In *Proc. of ICASSP-97, Munich, Germany,* 1997, pp. 983–987.
21. van Vuuren, S., *Speaker Recognition in a Time-Feature Space.* Ph.D. thesis, Oregon Graduate Institute of Science and Technology, Portland, OR, 1999.
22. van Vuuren, S., and Hermansky, H., !MESS, a modular efficient speaker verification system. In *Proc. of Speaker Recognition and its Commercial and Forensic Applications, France,* 1998, pp. 198–201.