Robust Text-Independent Speaker Identification over Telephone Channels

Hema A. Murthy, Françoise Beaufays, Larry P. Heck, Mitchel Weintraub[1]

EDICS Number: SA 1.8.1, manuscript number SA–594

(Revised version – Dec. 1997)

Corresponding Author: Françoise Beaufays
Address: Room EJ 129, SRI International,
333 Ravenswood Ave, Menlo Park, CA 94025, USA
e-mail: francois@speech.sri.com
Phone: (650) 859 5698
Fax: (650) 859 5984

Abstract

This paper addresses the issue of closed-set text-independent speaker identification from samples of speech recorded over the telephone. It focuses on the effects of acoustic mismatches between training and testing data, and concentrates on two approaches: extracting features that are robust against channel variations, and transforming the speaker models to compensate for channel effects.

First, an experimental study shows that optimizing the front end processing of the speech signal can significantly improve speaker recognition performance. A new filterbank design is introduced to improve the robustness of the speech spectrum computation in the front-end unit. Next, a new feature based on spectral slopes is described. Its ability to discriminate between speakers is shown to be superior to that of the traditional cepstrum. This feature can be used alone or combined with the cepstrum.

The second part of the paper presents two model transformation methods that further reduce channel effects. These methods make use of a locally collected stereo database to estimate a speaker-independent variance transformation for each speech feature used by the classifier. The transformations constructed on this stereo database can then be applied to speaker models derived from other databases.

Combined, the methods developed in this paper resulted in a 38% relative improvement on the closed-set 30-second training 5-second testing condition of the NIST'95 Evaluation task, after cepstral mean removal.

[1] The authors are with the Speech Technology and Research Laboratory, SRI International, Menlo Park, CA. Hema A. Murthy was on leave from IIT, Madras, India during this work.

I. Introduction

In many applications of automatic speaker recognition (ASR), a communication channel separates the user from the recognition system. This can, for example, be a telephone channel (*e.g.* identity verification for banking transactions over the phone, voice recognition for smart voice mail systems), or some type of microphone (*e.g.* voice identification for building access, identification of specific speakers in multimedia recordings). In most cases, the communication channel is allowed to change between different calls to the system, and the data samples collected to train the speaker models are only representative of a small fraction of all the acoustic conditions that can be met during testing. The resulting acoustic mismatches between training and testing data greatly affect the performance of ASR systems.

In this paper, we focus on closed-set speaker recognition from data collected over the telephone. The problem posed by acoustic mismatches can be tackled at different levels. In this paper, we concentrate (1) on the extraction of robust speech features and, (2) on a transformation of the speaker models to reduce channel effects. Recently, acoustic mismatches have also addressed at the classification level, by estimating which type of telephone unit is used in the communication and by modifying the speaker classification algorithm accordingly[1], [2].

A. The Feature Extraction Problem

The exact factors in a speech signal that are responsible for speaker characteristics are not exactly known, but it is a fact that humans are able to distinguish among speakers based on their voices. Studies on inter-speaker variations and factors affecting voice quality have revealed that there are various parameters at both the segmental and suprasegmental levels that contribute to speaker variability [3], [4], [5], [6], [7]. Despite the fact that one cannot exactly quantify inter-speaker variability in terms of features, current speaker identification systems perform very well with clean speech. However, the performance of these systems can decrease significantly under certain acoustic conditions, such as noisy telephone lines [8].

In the last few years, much of the speaker identification research has been devoted to modeling issues (*e.g.* [9], [10], [11], [12], [13], [14]), and significant performance improvements have been reported from developing sophisticated speaker models. Comparatively fewer papers have addressed the equally important issue of robust feature extraction for the purpose of speaker identification. Many current speaker recognition systems rely on spectral-based features, in particular the mel-cepstrum. A notable exception is the work by Janowski, Quatieri and Reynolds [15], where a new set of features based on amplitude and frequency modulation of speech formants and high-resolution measurement of fundamental frequency is used in addition to the standard filterbank-based cepstrum to perform speaker identification over a degraded channel. A drawback of this approach is that it requires an estimate of potential formant locations, which can be problematic. In addition, the performance of the system improves only when the new features are combined with the traditional mel-cepstrum.

In this paper, we first show experimentally that speaker recognition performance strongly depends on the front-end unit that preprocesses the speech signal. We demonstrate that the front end can be optimized to consistently and significantly improve the system performance. We also describe a new filterbank design that improves the robustness of the speech spectrum computation. We then derive a new feature based on spectral slopes that may be used either individually or in combination with the mel-cepstrum. Numerical results are provided to illustrate the performance gain brought by these algorithms.

B. The Model Transformation Problem

The second part of this work aims at developing transformation algorithms that render the speaker models more robust to acoustic mismatches.

Many ASR systems rely on cepstral mean subtraction (CMS) [16] to compensate for channel effects [17], [11]. It is well-known, however, that channel mismatches can still be a significant source of errors after CMS. Preliminary experiments reported in Section III-B confirm this point. For this reason, more sophisticated cepstrum transformation methods have been proposed in the literature. In [18], [19], cepstral compensation vectors are derived from a stereo database and applied to the training data to adjust for environmental changes. The compensation vectors depend either on the SNR or on the phonetic identity of the frames. In [21], an affine transformation of the cepstral vectors is estimated from a stereo portion of the database under study, and then applied to the training data.

The effect of transmission channels on speech has also been addressed in the context of speech recognition, where acoustic mismatches increase the confusability between phones and lead to word recognition errors. However, few of the algorithms developed for speech recognition can be readily applied to the problem of speaker recognition.

For example, adaptation algorithms that adjust the features or the models to better represent the test data (*e.g.* [22], [23], [24], [25]) are hard to use in speaker recognition: if the speaker models are adapted with the test data, they all eventually converge toward the same model, and the speaker discrimination capability is lost. Other speech recognition algorithms have addressed the mismatch issue by assuming that a priori knowledge about the mismatch is available: some algorithms require stereo data representing both conditions (*e.g.* [26], [27]), others need samples of similar sounds across different channels (*e.g.* [28]). These approaches are hard to implement in speaker recognition because of the practical difficulty of requiring each speaker to record large amounts of speech over multiple channels. In the case of telephone speech, this problem could be alleviated by clustering the channels in two or three categories. For example, a natural choice would be carbon button versus electret handsets [29]. The ASR system would then require a handset detector in order to select one or the other transformation. In this work, however, we prefer to assume that no a priori knowledge about the mismatch is provided or extracted from the speech waveform, and we show that significant improvement can be achieved without such knowledge.

The technique we propose compensates for channel mismatches by transforming the speaker models. It makes use of an auxiliary database containing stereo recordings to compute what we refer to as a *synthetic variance distribution*. This distribution can then be used to derive a transformation that is applied to the variances of speaker models built with training data from other databases. Two such transformations are proposed. They essentially increase the variances of the speaker models by an appropriate amount to render the speaker models more robust to channel effects. These transformations can be applied to different speech features, and have been tested both with cepstrum-based models and with models based on the new feature described in the paper.

The experiments reported in this paper are concerned with "closed-set" speaker recognition, that is with the problem of recognizing a speaker among N known speakers. In the last section of the paper, we show that the methods developed for closed-set ID also extend to "open-set" speaker recognition, that is to the problem of determining whether a test speaker belongs to the training set, and of identifying him if he does.

II. Databases Used in This Study

The focus of our effort is to ensure that the algorithms we develop are general and work on different telephone databases. To establish that this is the case, we used several corpora in this study. These corpora span different mismatch conditions, contain different amounts of training and testing data, were collected by different institutions, and illustrate different kinds of speech, from read digits to unconstrained conversational speech.

A. The Switchboard NIST'95 Evaluation Database (NIST95)

This database is a subset of the Switchboard corpus [30], collected by the Linguistic Data Consortium. It consists of conversational telephone speech and was used as a benchmark for the NIST'95 (National Institute of Standards and Technology) Evaluations [31]. The database consists of 26 speakers, 13 male and 13 female. We experimented only with the 30-second training, 5-second testing condition. The training data consists of three 10-second segments taken from different conversations. The test data consists of 36 5-second segments per speaker. The 36 segments were taken from 6 different conversations. About 50% of the conversations from which the test data were extracted were conducted on the same telephone handset as the training conversation. We used this database as a benchmark throughout our work.

B. The Switchboard-45 Database (SB45)

We assembled the Switchboard-45 database from the Switchboard corpus by choosing 45 speakers, 19 male and 26 female, who were not in the NIST95 database. The training data varies from 20 to 40 seconds per speaker and the testing data consists, for each of the speakers, of approximately 100 segments of lengths greater than 2 seconds. The test data were extracted from different conversations than the training data.

C. The SRI-Digits Database

The SRI-digits database contains the voices of 10 male SRI employees. The text of the data consists only of spoken digits. The data were collected from 20 to 23 telephone handsets, all connected to the same internal telephone line. Six sets of three calls were made from each handset. The three calls in each set contain repetitions of the same number.

This database provides a lot of flexibility for designing experiments to test our system on different training environments. We set up four different training conditions, namely, train on one handset (SRI-1), train on all handsets (SRI-2), and train on multiple handsets (SRI-3, SRI-4). The test data were kept identical throughout the experiments. For each speaker, 180 test segments are available. Except for the dataset SRI-2, the telephone units used in training were never used in testing.

D. The Stereo-ATIS Database

The Stereo-ATIS database contains read sentences concerning flight-related issues. The sentences were recorded simultaneously with a Sennheiser close-talking microphone and over a local telephone line. The database was collected at SRI and contains the voices of 13 male speakers. Each speaker read 10 sets of about 30 sentences. Each set of sentences was recorded with the same telephone line but with a different handset. Sentences are on average four seconds long. The amount of data used for training and testing was varied according to the experiments. Because it contains stereo speech, this database is ideally suited for controlled experiments. It was extensively used in the development of the channel compensation algorithms.

E. The NIST'96 Evaluation Database (NIST96)

This database is a subset of the Switchboard corpus. It contains the 156 male speakers to be recognized in the NIST'96 Evaluation task (target speakers). We consider only the following condition: The training data for each speaker consists of 2 minutes of speech extracted from a single conversation, the test data consists of a total of 1357 segments of 30 seconds each (8.7 segments per speaker, on average). The test segments were extracted from conversations recorded over telephone channels not seen in training. This is thus a highly mismatched database. It was used to check the performance of the model transformation method with a large pool of speakers.

III. Baseline System and Preliminary Experiments

A. Description of the Baseline System

The speaker recognition system used as a baseline for this work consists of a series of Gaussian mixture models (GMMs) modeling the voices of the speakers to be identified, along with a classifier that evaluates the likelihood of unknown speech segments with respect to these models. In closed-set problems (speaker identification), the classifier hypothesizes the identity of an unknown test speaker by determining which model maximizes the likelihood of the test utterance. In open-set problems, the classifier compares the likelihood scores to some threshold to reject the test segments that poorly match all the trained models, and otherwise hypothesizes speaker identities based on likelihood maximization.

In both training and testing, the speech waveforms are digitized and preprocessed by a front-end unit that extracts a set of N_c mel-frequency cepstral coefficients from each frame of data. The parameters that control the front-end processing (*e.g.* frequency range, filter shape, filter resolution) are set a priori but can be modified if desired. Cepstral mean subtraction can be applied (optionally) to each utterance to eliminate some of the spectral shaping occurring in the communication channel.

For each speaker, a GMM is built using the speaker's training data to estimate the prior probabilities, means, and variances of the Gaussians. The number of Gaussians, N_g, is the same for each speaker, and is chosen depending on the amount and nature of the training and testing data. The generation of a GMM starts with the choice of a random vector quantization (VQ) codebook. The codebook is then iterated on, using the Lloyd algorithm [32]. At each VQ iteration, the Gaussians containing the largest number of data points are split, and the Gaussians containing the fewest data points are eliminated. After convergence of the VQ codebook, Gaussians are fitted to the codewords, and their parameters are adjusted using a few expectation-maximization (EM) iterations [32].

When presented with a speech segment from an unknown speaker, the classifier scores all the sufficiently high energy frames of the segment against all speaker models, accumulating the log-likelihoods of the speech frames for each model. A hard decision is then made as to which model summed up the highest log-likelihood, and this model is hypothesized as belonging to the speaker who uttered the test segment.

B. Preliminary Experiments

Preliminary experiments were performed with the baseline system to measure the effect of channel mismatches and cepstral mean subtraction on speaker recognition error rates. These experiments are performed wit a 16-coefficient mel-cepstrum feature.

Using the Stereo-ATIS database described previously, a set of 64-Gaussian GMMs was built with 40 seconds of Sennheiser-recorded speech per speaker. The system was then tested with 4-second Sennheiser utterances (lines 3 and 4 in Table I) and with the telephone recordings of the same utterances (lines 1 and 2). For comparison, 10 sets of 64-Gaussian GMMs were built with speech recorded from 10 different telephone units, and the models were tested with speech from the same telephone units. The performances of these 10 matched telephone-telephone systems were averaged and reported in line 5 of Table I.

The results reported in the table show that although CMS reduces the error rate significantly in the mismatched system (from 33% to 16%), its error rate with CMS remains more than three times higher than that of the corresponding matched Sennheiser system (16% vs. 5%). Other researchers (*e.g.* [33], [34]) have reported similar results: CMS eliminates convolutional effects but it does not eliminate additive noise and does not take into account channel nonlinearities and nonstationarities.

In addition, CMS can eliminate some of the speaker characteristics as exemplified by the matched Sennheiser experiments (lines 3 and 4). This result, which may be attributed to the cancellation of vocal-tract information by the high-pass filtering in CMS, is to be expected also from techniques such as RASTA preprocessing [33].

Finally, comparing lines 3 and 5 in Table I confirms that, more than the presence of a telephone unit between the speaker and the ASR system, it is the possible mismatch between training and testing conditions that results in poor speaker-ID performance.

In another series of experiments with the Stereo-ATIS database, we measured the average distortion between Sennheiser-recorded cepstral coefficients and their telephone stereo recordings. The distortion measure for cepstral coefficient k, d_k is defined as

$$d_k = \frac{< (c_k^S - c_k^T)^2 >}{\sigma_k^S \sigma_k^T},$$

where c_k^S and c_k^T denote, respectively, the k^{th} cepstral coefficients of a frame of Sennheiser-recorded data and of its telephone stereo-recording; and σ_k^S and σ_k^T denote the standard deviations of the Sennheiser and telephone cepstral coefficients. The average $< . >$ is taken over all the telephone units and all the speakers in the database, and is estimated from all the frames of several sentences of each speaker-telephone combination. Fig. 1 shows the average distortion d_k versus the cepstral coefficient index k, with and without cepstral mean subtraction. Again, cepstral mean subtraction helps decreasing the effects of the channel, although the distortion remains significant after CMS. The figure also shows that the channel effects are more noticeable on higher-order cepstral coefficients. This may be due to the fact that the overal speech energies in these coefficients are lower than those in lower-order cepstral coefficients, and that noise effects are therefore relatively more important.

Because this work focuses on speaker recognition under mismatched conditions, CMS was systematically applied throughout the paper (unless otherwise specified) as a first step to eliminate channel effects.

IV. Feature Extraction for Speaker Identification

In this section, we discuss some issues regarding the extraction of features for speaker recognition. We show that the performance of an ASR system depends strongly on the parameters describing the front-end unit that processes the incoming speech. To make the front end more robust over a large range of parameters, we redefine the filterbank based on which the cepstrum feature is computed. We then demonstrate experimentally that a large performance improvement can be obtained by optimizing the front-end parameters. We report detailed experimental results and suggest intuitive explanations wherever possible.

We then introduce a new feature: the spectral slope. We show that the spectral slope (after re-optimization of the front-end parameters) discriminates better between speakers than the cepstrum. In addition, we argue that the two features contain relatively orthogonal information, and we show that combining them further improves the system performance.

A. Description of the Baseline Front End

The baseline front end used in this work first transforms the speech signal to the frequency domain via a fast Fourier transform (FFT). The frequency scale is then warped according to the mel-scale to give a higher resolution at low frequencies and a lower resolution at high frequencies. Specifically, we implemented the bilinear transformation proposed by Acero[20],

$$\omega_{\mathrm{new}} = \omega + 2\ \mathrm{atan}\frac{F_w sin\omega}{1 - F_w cos\omega},$$

where the constant $F_w \in [0,\ 1]$ controls the amount of warping. The frequency scale is then multiplied by a bank of N_f filters whose center frequencies are uniformly distributed in the interval $[\mathrm{Min_f},\ \mathrm{Max_f}]$, along the warped frequency axis. The width of each of these filters ranges from the center frequency of the previous filter to the center frequency of the next filter. The filter shape is trapezoidal, and can vary all the way from triangular to rectangular. The shape of a particular set of filters is encoded in a constant T_r that measures the ratio of the small to the large side of the trapezoid ($T_r = 0$ means triangular filters, $T_r = 1$ means rectangular filters). The filterbank energies are then computed by integrating the energy in each filter, and a discrete cosine transform (DCT) is used to transform the filterbank log-energies into cepstral coefficients. Cepstral mean subtraction is applied to each training and testing utterance.

B. Perceptually Motivated Filterbank Design

Because the bandwidths of the mel-warped filters are chosen based on the number of filters, the filterbank energy estimates can be very poor at certain frequencies when the frequency scale is warped, especially if the number of filters is large. Ideally, one should nonuniformly sample the Fourier transform during the computation of the FFT to compensate for the warped frequency scale. Alternatively, one can make the bandwidth of each filter a function of frequency (as opposed to a function of the distance between center frequencies of adjacent filters as in the baseline design). The question is how this function should be chosen.

In [35], Klatt proposed a bank of 30 filters based on auditory perception criteria. Because Klatt's parameters are derived for a fixed filterbank size, his design could not be ported directly to our system. Instead, we approximated Klatt's coefficients with the function given in Table II. To provide more flexibility to the system, an additional parameter, the bandwidth scale B_s, was introduced in the front end to uniformly scale the filter bandwidths, i.e.

$$Bw_{\text{new}} = B_s \; Bw,$$

where Bw is the bandwidth specified in Table II.

C. Optimization of the Front-End Parameters for the Filterbank-Based Cepstrum

A number of parameters affect the computation of the cepstrum. These are:

N_c	- number of cepstral coefficients
N_f	- number of filters in the filterbank
Min_f- Max_f	- effective voice bandwidth
F_w	- frequency warping constant: -1.0 to +1.0
T_r	- shape of the filters: 1.0 (rectangle), 0.0 to 1.0 (trapezoid), 0.0 (triangle)
B_s	- scale factor for the bandwidth: 0.0 to 1.0

From a signal processing perspective, N_c defines the resolution that is required in the cepstral domain, N_f defines the resolution that is required in the frequency domain, Min_f and Max_f define the effective voice bandwidth, F_w defines the resolution at different frequencies, and T_r defines the shape of the filters. But how do these parameters affect the classifier performance?

A series of preliminary experiments showed that the values of most front-end parameters had a large impact on the classifier performance. For example, varying the filter shape while keeping the other parameters constant would make the error rate vary between 26% and 29%. Other researchers observed similar results (*e.g.* [36] describes an experimental study on the influence of filterbank design and parameters on isolated word recognition).

Since it is not clear how each individual front-end parameter affects the classification error rate, we performed an extensive optimization of all the front-end parameters, using the NIST95 database. To ensure that the optimized parameters were not specific to that database, we collected the sets of parameters that gave the best performance on NIST95 and tested them on a series of other databases. The front-end parameters that resulted in the best performance across all the databases were then chosen as the new front-end for the speaker-identification system using that feature.

The optimization method we used is based on successive line searches. At each iteration, all the parameters but one are held constant, and the speaker classification error rate is evaluated for different values of the remaining parameter (line search). The value that leads to the lowest error rate is retained. The optimized parameter is then fixed, and the next parameter is optimized. The procedure continues until the error rate reaches a local minimum. The front-end parameters used to initialize the optimization procedure (see Table III) resulted from partial optimizations done previously in our laboratory. These initial parameters were those used in the baseline system.

Table IV gives the list of experiments that were conducted to evaluate the performance of the system for different front-end parameters. The filterbank used in these experiments is that described in the previous section.

Fig. 2 illustrates the performance of the system for different parameter values. Each figure shows the speaker-ID error rate as a function of one parameter. For each value of this parameter, a series of experiments was performed by varying the other parameters and measuring the resulting speaker-ID error rates. The lowest error rate over each set of experiments was retained and plotted against the parameter of interest. From Fig. 2(a), we observe that the performance is best for B_s ranging from 0.8 to 1.0. It was also observed in the experiments on B_s that the performance was uniformly poor for $F_w > 0.2$ and $T_r < 0.5$ (not displayed in Fig. 2).

The overall performance of the system uniformly improves as the number of cepstral coefficients increases, up to 17. Beyond $N_c = 17$, the error rate begins to increase (Fig. 2(b)). It is likely that for low orders of cepstral coefficients, speaker information dominates in the representation but, as the number of cepstral coefficients increases, the channel information begins to dominate (see Section III).

Fig. 2(c) shows the result of varying N_f with N_c fixed at 17. Although the accuracy of the estimates of the cepstral coefficients depends upon the number of filters used in the computation of the filterbank energy coefficients, the error rate does not vary significantly with N_f. This may be because the filter bandwidths are independent of the number of filters; adding more filters is thus equivalent to interpolating the filterbank log-energies and does not add to the resolution of the spectrum.

From Fig. 2(d), we observe that the performance of the system is quite sensitive to Min_f and Max_f. The error rate is uniformly high when the effective voice bandwidth is decreased significantly.

Table V gives the parameters of the best front end using the cepstrum feature, for the NIST95 database. Comparing Table III and V, we see that the improvement in error rate due to the parameter optimization and to the modification of the filterbank computation is 25.6% relative.

Since the performance of the system varies significantly with the choice of front-end parameters, the next test that must be performed is to determine whether this performance gain holds up on different databases. From the set of experiments performed in Table IV, approximately 50 of the best systems were chosen and the performance of these systems was evaluated on the SRI-digits and Switchboard-45 databases. The parameter values that resulted in the best performance varied across the different databases. The system that resulted in the best average performance was chosen as the new front end for the speaker-ID system using the cepstrum as a feature. The front-end parameters for this system happen to be identical to those obtained for NIST95 alone (Table V).

D. Filterbank-Based Spectral Slope Along Frequency

We have discussed in a previous section how we computed the filterbank-based cepstrum. The information contained in the cepstrum corresponds to the overall shape of the spectrum. It is likely to be dominated by the first formant since the first formant has the highest energy, due to the effect of the glottal roll-off. It is well known that formants and their transitions are very important for the perception of speech. In psychophysical studies performed by Klatt [37], it was observed that when formant locations are changed, the sounds perceived by listeners are different from what was intended. The same study shows that humans perceive the same sound when the relative amplitudes of the formants are modified in different instantiations of the sound.

Although various algorithms have been developed to estimate formant frequency locations in running speech (*e.g.* [38], [39], [40]), the formant-extraction problem is nontrivial. Machines tend to make gross errors in estimating formant locations: spurious peaks are introduced and true peaks are often missed. We therefore looked for a new measure that would emphasize the locations of formants without actually estimating them.

Filterbank-based spectral slope is a metric that can do this. When comparing the slopes of two spectra of the same sound, the amplitude differences are not captured, while the locations and bandwidths of resonances are captured. The spectral slope can also be related to the shape of the glottal pulse. If we assume a source-system model for speech production, the spectra corresponding to the system ride on top of the spectra corresponding to the source. Even if the peak locations are the same for different speakers, the slope information can give information about the tilt in the spectrum, the spectral tilt being related to the shape of the glottal pulse.

A spectral slope metric was suggested by Klatt [37] and used by Hanson and Wakita [41] for isolated word recognition. In the latter study, the slope is computed indirectly, using the relationship between the derivative of the spectrum and the weighted cepstrum. This principle can be applied to the filterbank-based cepstra only when the number of filters is infinite (and nonoverlapped) and the number of cepstral coefficients is infinite. Neither of these conditions is true in practice. We therefore propose a technique based on the metric suggested by Klatt [37] but where the slopes are computed differently.

As with the cepstrum feature, the speech signal is transformed to the frequency domain via an FFT, and the frequency scale is warped. The spectrum is then multiplied by a bank of filters similar to that used for the baseline cepstrum. In a first implementation, the spectral slope was computed as the difference between the log-energies of two consecutive filters. The best performance with this system on the NIST95 database resulted in a 31.7% error rate.

This system was then improved in three ways. First, the original filterbank was replaced with the perceptual filterbank of Section IV-B. Then, CMS was introduced in the filterbank slope computation to reduce channel effects. This was done by taking the DCT of the filterbank log-energies, eliminating the first cepstral coefficient, c_0, and computing the inverse DCT of the remaining cepstral coefficients. The spectral slopes were then computed from the transformed filterbank. Last, the slope computation was made more robust to small variations in the filterbank log-energies by using a 3-point regression technique instead of a simple difference between adjacent filters. This new system was optimized as detailed below.

E. Optimization of the Front-End Parameters for the Spectral Slope

The front-end parameters were reoptimized for the spectral slope feature and tested on all the databases. In the context of the spectral slope, we also found that the performance of the system varied significantly with the choice of front-end parameter values. Again, successive line searches were performed on the NIST95 database (Table VI) to select the best front end. Fig. 3 shows the optimization of the front end for different parameters. From the experiments on B_s (Fig. 3(a)) we notice that the performance of the system is uniformly good for a choice of B_s between 0.6 and 1.0.

Fig. 3(b) shows the results of the optimization of N_f. From Fig. 3(b), it appears that the system with 28 filters works best for $B_s = 1.0$. The parameters of the system that worked best on the NIST95 database are given in Table VII. We did not perform experiments on Min_f and Max_f since the results on the cepstrum did more or less indicate that the entire voice bandwidth is important. We tested the performance of approximately 50 of the best systems on the SRI-digits and Switchboard-45 databases. The system that resulted in the best average performance was identical to the best NIST95 system.

F. Discussion

The experiments we have described indicate that the performance of the speaker-ID system fluctuates significantly with the choice of the front-end parameters. This fluctuation could be due to one of two reasons: (1) the features are very sensitive to the front-end parameters, and (2) the models generated are sensitive to small changes in parameter values. It is possible that the probability density functions used to represent the features are not Gaussian. However, given that each element of the feature is represented by a Gaussian-mixture density function, a poor fit between the model and the data is unlikely. The variation across databases should thus be attributed to the variation in channel characteristics across the different databases and to the sensitivity of the front-end parameters to the channel characteristics. The new features with the new front end are still sensitive to channel variations. In Section V we address this issue from a modeling point of view, and show that the new features along with modified models can significantly improve the performance of the system.

G. Combining Different Features

In most of the experiments we performed on speaker identification, using of the cepstrum or the spectral slope resulted in similar performance. If the two features carry complementary information, their combination can be expected to perform better than either feature alone. To verify this hypothesis, we combined the two features by taking, for each test utterance, a simple average of the normalized log-likelihoods of the observations obtained for each feature individually. The normalization factor for each feature is simply the length of the feature vector. This prevents the feature vector with the highest number of components from dominating the overall score.

The performances of the systems using each of the features individually and the combined features across all databases are shown in Table VIII. Note that the combined systems did work uniformly better than either feature alone, except on SRI-3 where the spectral slope could not benefit from the additional information brought by the cepstrum.

H. Extension to Speech Recognition

The techniques described in the previous sections were tested on a *speech recognition* task. We determined two new sets of front-end parameters for speech recognition by Viterbi aligning the transcriptions of a few hours of Switchboard speech, modeling each context-independent phone with a GMM, and finding the front-end parameters that minimized the phone classification error rate for the cepstrum and spectral slope features. These front ends were then used to perform a speech recognition experiment on the 1995 development set of the Large Vocabulary Continuous Speech Recognition (LVCSR) Evaluation on the Spanish Callhome database[42]. Cepstral mean subtraction was applied at the sentence level, in all the experiments. Results are summarized in Table IX.

Table IX shows that optimizing the front-end parameters brought a 3.6% absolute reduction in word error rate over a state-of-the-art speech recognizer, which is a significant improvement given the difficulty of the task. However, the improvement brought by the spectral slope in speaker recognition problems *does not carry over* to the context of speech recognition. This confirms a posteriori that the spectral slope conveys information that is specific to the speaker, *e.g.* the glottal roll-off (see Section IV-D), rather than to the speech.

V. Model-Based Channel Compensation Methods

In the previous section, we addressed the problem of acoustic mismatches between training and testing data from a feature-extraction viewpoint. In this section, we propose a model-based channel compensation method that aims at reducing remaining channel effects. The framework for which this method was developed assumes that – as in many speaker identification applications – training data are collected from only a few telephone units, whereas the system is expected to recognize the speaker's voice from many other handset-line combinations.

In terms of Gaussian mixture modeling, a change in acoustic environment translates into a modification of the means and variances of the clusters of features representing the speaker's voice. As a consequence, the speaker's test data are not well modeled by the Gaussians built to fit the training data, and speaker misidentifications are likely to occur.

Deriving model transformations that counteract these parameter changes is made difficult by the fact that collecting data from many different telephone lines for each speaker in the database is often impractical. Whereas in speech recognition a large variety of acoustic environments can be obtained by pooling speech from different speakers using different units, in speaker recognition each model must be trained with data from only one speaker. In this context, a more practical approach is to collect multi telephone data from a few patient speakers, analyze these data, and try to apply the resulting observations to other databases.

The method we propose essentially performs channel *compensation*, as opposed to channel *adaptation*. It aims at rendering the speaker models more robust to channel mismatches by appropriately increasing their variances while keeping their means unchanged. The variance increases are different along each cepstral coefficient. They are meant to account for the unknown shifts in the means occurring with the features when the channel changes, as well as for possible variance modifications. Fig. 4 illustrates this conceptually in a two-dimensional feature space. If G1 is a cluster of features observed on the training data collected from a given telephone unit, the same speech frames transmitted by another unit might look like G2 or G3 or G4. Since our baseline system uses Gaussian mixture models, we can think of G1 as one Gaussian of a speaker's GMM. The exact mean and variance changes from G1 to G2, G3, or G4 are generally unknown at the time of testing. Instead of trying to estimate them from the data, we replace G1 with G, a Gaussian that "covers" the possible regions where we may expect the data to lie when transmitted by different telephone lines. The variances of the G clusters of all the speaker models form what we refer to as a *synthetic variance distribution*. This variance distribution can then be used to derive variance transformations for other databases.

As argued in the next section, this approach can also to some extent compensate for two other factors: the typically limited amount of training data and the limited size of the speaker models.

A. Amount of Training Data and Model Size

In matched conditions, the performance of a speaker identification system largely depends on the amount of training data available: the more data there are, the better the speaker's voice can be modeled, and the lower the error rate is. This observation also holds for mismatched systems as illustrated in Fig. 5. In this experiment, we used increasing amounts of Sennheiser data from the Stereo-ATIS database to build four GMMs having, respectively, 64, 128, 256, and 512 Gaussians. We then tested the models with two sets of data: one contained Sennheiser utterances, the other contained the stereo recordings of the same sentences, recorded from various telephone units. The two test sets were kept unchanged as the amount of training data increased. Fig. 5 compares the performance of the matched and mismatched systems and shows that even if mismatched with the test data, more training data significantly decreases the speaker-ID error rate. It also shows that larger amounts of training data allow models with more Gaussians to outperform smaller models.

The amount of data used to build a GMM and its number of Gaussians is directly reflected by the variance distribution of the Gaussians. For illustrative purposes, we computed, for each GMM built in the previous experiment, the average along each cepstral coefficient of the variances of the Gaussians in the GMM. The averages along c_2 are plotted in Fig. 6. The figure shows that, for a given amount of data, the Gaussians of large GMMs have lower variances since they model fewer data points. It also shows that, for a given model size, the average variance increases with the amount of training data. This occurs because the EM algorithm effectively tries to minimize the variances of the Gaussians in the model, which can better be achieved when there are fewer data points per Gaussian.

This observation suggests that artificially increasing the variances of GMMs may be useful to compensate for the lack of training data, and to allow larger models to be built. We will see this assumption verified in our experiments.

B. Synthetic Variance Distribution

Using the Stereo-ATIS database, which (see Section II) contains Sennheiser-recorded speech and telephone-transmitted speech recorded in stereo mode, a synthetic variance distribution can be computed as illustrated in Fig. 4. The Sennheiser utterances of the database are used to build the G1 clusters, and their telephone stereo recordings are used to estimate the variances of the G clusters. Because lower-order cepstral coefficients typically have a larger dynamic range than higher-order coefficients, the variance distribution is estimated separately for each direction of the cepstral feature space.

The algorithm for computing the synthetic variance distribution can be summarized as follows:

1. Set apart a few Sennheiser sentences from each speaker and build with them a set of N_g-Gaussian GMMs that will be used as frame classifiers.
2. For each speaker in the database:
   (a) Use the speaker's GMM to label each frame of the speaker's remaining Sennheiser data with the index of the Gaussian that maximizes its log-likelihood, that is, classify the Sennheiser frames into N_g clusters.
   (b) For each Gaussian in the GMM (for each cluster):
      i. Compute the mean, $\boldsymbol{\mu}_S$, and the variance, $\boldsymbol{\sigma}^2_S$, of the Sennheiser frames clustered by this Gaussian.
      ii. Compute the variance, $\boldsymbol{\sigma}^2_T$, of the stereo recordings of these frames. These stereo recordings comprise frames recorded on various telephone units (10 in total in Stereo-ATIS). To compensate for the shift in the means occurring between the Sennheiser and telephone data, the variance $\boldsymbol{\sigma}^2_T$ is computed with respect to the mean $\boldsymbol{\mu}_S$ of the Sennheiser frames rather than with respect to the mean of the telephone frames, $\boldsymbol{\mu}_T$.

   The variances, $\boldsymbol{\sigma}^2_T$, form the desired synthetic variance distribution.

We used boldface symbols for the means and variances to emphasize that these are vectors of N_c (the number of cepstral coefficients) elements. The synthetic variance distribution is thus N_c-dimensional.

We built such a synthetic variance distribution from the Stereo-ATIS database, keeping 30 sentences from each of the 13 speakers in the database to train a set of 64-Gaussian GMM classifiers, and using the other 270 sentences per speaker to derive the synthetic variance distribution. The feature used was the 17-dimensional cepstrum.

Fig. 7 displays pairs of variances ($\boldsymbol{\sigma}^2_T$, $\boldsymbol{\sigma}^2_S$) computed along two different cepstral coefficients, c_1 and c_{17}. Each plot contains 13 × 64 points (the number of speakers in the database times the number of Gaussians in the speaker GMM). The data points in each plot were normalized to have zero mean and unit variance.

Fig. 7 shows that (1) as we expected, most of the telephone variances are larger than the corresponding Sennheiser variances, and (2) the variances along c_{17} show more dispersion than those along c_1. This is not unexpected since we have observed in Section III that higher-order cepstral coefficients are more sensitive to channel effects.

C. Affine Transformation of the Variances

In first approximation, the data points in each plot of Fig. 7 can be fitted with a straight line. The coefficients of these straight lines define an *affine transformation* of the Sennheiser variances onto the telephone variances. Speaker models trained from databases containing speech collected from a single acoustic environment (single handset, single telephone line) can benefit from this transformation to modify their variances and increase their acoustic coverage.

Expressing the affine transformation as $y = m_i x + t_i$, where i refers to the cepstral coefficient c_i, and x and y represent, respectively, the variance of a cluster of Sennheiser frames and the variance of the corresponding telephone frames, the parameters m_i and t_i can be estimated from the data, using a least mean squares fitting:

$$\begin{aligned} m_i &= \frac{<\boldsymbol{\sigma}^2_{T,p,j}(i)><\boldsymbol{\sigma}^2_{S,p,j}(i)> - <\boldsymbol{\sigma}^2_{S,p,j}(i)\boldsymbol{\sigma}^2_{T,p,j}(i)>}{(<\boldsymbol{\sigma}^2_{S,p,j}(i)>)^2 - <(\boldsymbol{\sigma}^2_{S,p,j}(i))^2>}, \\ t_i &= <\boldsymbol{\sigma}^2_{T,p,j}(i)> - m_i <\boldsymbol{\sigma}^2_{S,p,j}(i)>, \end{aligned}$$

where $\boldsymbol{\sigma}^2_{S,p,j}(i)$ and $\boldsymbol{\sigma}^2_{T,p,j}(i)$ denote, respectively, the variance of the Sennheiser and telephone data for the j^{th} Gaussian of speaker p's model, along cepstral coefficient c_i, and where $< \cdot >$ indicates the average over p and j, *i.e.* over all the Gaussians of all speakers.

The variance transformation equations are then described by

$$\boldsymbol{\sigma}^2_{\text{tfmed},q,l}(i) \stackrel{\triangle}{=} m_i\, \boldsymbol{\sigma}^2_{q,l}(i) + t_i,$$

where $\boldsymbol{\sigma}^2_{q,l}(i)$ represents the variance to be transformed (more specifically the variance of the l^{th} Gaussian of speaker q's model, along c_i), and $\boldsymbol{\sigma}^2_{\text{tfmed},q,l}(i)$ represents the same variance after transformation.

C.1 Results of Experiments

The affine transformation developed on Stereo-ATIS was applied to the SRI-digits database described in section II. Gaussian mixture models were trained with 1 minute of speech collected from one telephone line (line 1 or line 2) and tested with multi line data. Table X compares the error rates with and without variance transformation for different model sizes. (The last two lines in Table X will be explained in Section V-D). Although the transformations were derived from Stereo-ATIS, they significantly improved the performance on this new database. As we argued in Section V-A, increasing the model variances also allowed us to increase the model sizes.

One problem with the affine transformation method is that it implicitly assumes that training data are provided by a single acoustic environment. If, instead, training data are collected from a few different telephone lines, one might expect these data to cover more of the feature space and the reduced mismatch to require a different variance transformation. As stereo data are hard to collect from two or more telephone units simultaneously, another method must be developed to deal with this situation.

D. Translation to a Fixed Target

For analysis purposes, the affine transformation can be simplified into a scaling part modeled by the slope m_i or a translation part modeled by the offset t_i. Table X shows the error rates obtained by setting the slope m_i to one and estimating the offset using the least means squares, and the error rates obtained by setting the offset to zero and estimating only the scaling part of the transformation. The table shows clearly that the most significant part of the transformation is due to its additive offset component. This can be justified intuitively: in first approximation, the speech coming out of a telephone line can be represented in the cepstral domain as a random process resulting from the sum of a clean speech contribution and a channel effect. Since the signals are additive, so are their variances. Thus, the translation term in the variance transformation corresponds to an estimate of the average channel variance. The full affine transformation, with its scaling term, refines this model by taking into account nonadditive effects.

The *translation to a fixed target* method takes advantage of this observation to simplify and generalize the variance transformation and allow it to deal with multi line training and, as a by-product, to compensate for limited amounts of training data.

In this method, the synthetic variance distribution is seen as a "fixed target" to be reached by the variances of the speaker models. The variances of the speaker models are translated by an amount such as to make their mean equal to the mean of the synthetic variance distribution. Mathematically, the transformation can be described as

$$\boldsymbol{\sigma}^2_{\mathrm{tfmed},q,l}(i) \stackrel{\triangle}{=} \boldsymbol{\sigma}^2_{q,l}(i) + t_i,$$

where

$$t_i = < \boldsymbol{\sigma}^2_{T,p,j}(i) > - < \boldsymbol{\sigma}^2_{q,l}(i) > .$$

Provided that the synthetic variance distribution is computed with a large amount of training data (*i.e.* large enough to reach the asymptote in Fig. 6), the translation term t_i also corrects the speaker model variances, $\boldsymbol{\sigma}^2_{q,l}(i)$, for being underestimated because of a lack of training data.

In addition to its capability of compensating for small amounts of training data, this method extends easily to training conditions including more than one line since it does not make any assumption about the training conditions (as opposed to the affine transformation method).

D.1 Results of Experiments

The affine transformation and fixed target translation were compared in a set of experiments performed on the SRI-digits database, with one- and two-line training conditions. The fixed target translation method consistently outperformed the affine transformation method. Results are summarized in Table XI.

Next, a series of experiments was performed to check that the improvement brought by the fixed target translation holds for larger sets of speakers. We used to this effect the NIST96 database (see Section II), which contains mismatched data from 150 male speakers. Speaker-ID experiments were performed with 10, 25, 50, 100, and 150 speakers, with and without variance transformation. Whenever possible, we averaged the outcomes of several experiments to improve the precision of the error rate estimates. (For example, 15 10-speaker experiments were averaged to obtain the 10-speaker error rate. For the 25-, 50-, 100-, and 150-speaker error rates, respectively 6, 3, 1, and 1 experiments were performed and averaged. This way, all the training and testing sentences were used exactly once to determine the error rate for each number of speakers.) The results of these experiments are displayed in Fig. 8. The figure shows that the improvement brought by the variance transformation is essentially independent of the number of speakers. Transforming the model variances improves the match between the speaker models and the test data, irrespectively of the number of speakers to be in the database.

E. Variance Transformation with Multiple Features

So far, we assumed that the cepstrum was the only feature used for speaker identification. The following experiment shows the performance on the NIST95 database of a system combining the optimized features from Section IV with the fixed target variance transformation. Two

64-Gaussian GMMs were built from the cepstra and spectral slopes of the NIST95 training data, using the optimized front ends summarized in Tables V and VII. Two sets of variance transformations were computed for the same features and the same front ends, with Stereo-ATIS data. The transformations were applied to the NIST95 GMMs, and testing was done as described in Section IV-G, that is, by maximizing the sum of the normalized likelihoods of the two classifiers. The results, summarized in Table XII, show that the combined system reduced the error rate from 24.89% to 20.83%, a 16.31% relative error rate reduction (note that the baseline for this experiment, 24.89% error rate, assumes that the front end is already optimized).

F. Extension to Open-set Speaker Recognition

All the results presented in this paper were for closed-set speaker identification. Another important problem is that of open-set speaker recognition, where "target" speakers are to be identified and "imposter" speakers are to be rejected. Open-set speaker recognition involves many issues that are beyond the scope of this work, however we would like to close this paper with the results of an experiment we made on open-set recognition, and which shows that the performance improvement that we observed on closed-set identification holds up in the case of open-set speaker recognition. This experiment was conducted on NIST95 and extends Table XII to the case of open-set recognition. The target speakers for this experiment were the 26 speakers from the closed-set experiment. To these, 80 imposter speakers were added. The target speakers were modeled with 64-Gaussian models. Two speaker-independent background models were built (one computed from the cepstrum feature, the other from the cepstral slope feature) with data from the SB45 database. These models had, respectively, 2400 and 800 Gaussians. The likelihood scores produced by the target models where normalized by the likelihood scores from the speaker-independent background model, and likelihood maximization combined with a rejection threshold was used to identify or reject test utterances. The results of this experiment are summarized in Table XIII, where properties such as the closed-set error rate, the miss rate with a 3% false alarm rate, and the false acceptance rates with 10% and 1% miss rates are reported. The combination of the modified features along with the variance transformations significantly improved all the criteria that we evaluated.

VI. Conclusion

We have attempted to compensate for channel effects in speaker identification by optimizing the front-end parameters, modifying our filterbank computation, introducing a new feature, and transforming the speaker models. Although it has been shown that significant improvements are obtained, we have only scratched the surface. In the context of feature extraction, the performance gain is obtained by using only system features. There is a need to develop robust source-feature extraction algorithms. In the context of model transformation, the fixed-target compensation algorithm has resulted in a significant performance gain, but it has certainly not completely compensated for channel effects. This approach should be extended to speaker-dependent and microphone-dependent transformations, which we expect to give further improvements.

Acknowledgments

The authors wish to thank Leonardo Neumeyer, Ze'ev Rivlin, and Ananth Sankar from SRI International, Richard McGowan who was visiting SRI from the Haskins Laboratories during this work, and Mike Hochberg from Nuance Communications for many insightful discussions and comments.

References

[1] D. A. Reynolds, "MIT Lincoln Laboratory site presentation", *NIST Speaker Recognition Workshop*, Linthicum Heights, Maryland, 27 March 1996.

[2] L. P. Heck and M. W. Weintraub, "Handset-dependent background models for robust text-independent speaker recognition", to appear in *Proc. ICASSP-97*, Munich, Germany, April 1997.

[3] G. Doddington, "Speaker recognition–identifying people by their voices," *Proc. IEEE*, vol. 73, no. 11, November 1985.

[4] G. Fant, A. Kruckenberg and L. Nord, "Prosodic and segmental speaker variations," *Speech Communication*, vol. 10, pp. 521-531, 1991.

[5] D. H. Klatt and L. C. Klatt, "Analysis, synthesis and perception of voice quality variations among female and male talkers," *J. Acoust. Soc. Amer.*, vol. 87, no. 2, pp. 820-857, 1990.

[6] D. G. Childers and C. K. Lee, "Voice quality factors, analysis, synthesis and perception," *J. Acoust. Soc. Amer.*, vol. 90, no. 5, pp. 2394-2410, 1991.

[7] M. Narendranath, H. A. Murthy, B. Yegnanarayana and S. Rajendran, "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, pp. 207-216, 1995.

[8] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Proc. ESCA Workshop on Automatic Speaker Recognition*, pp. 27-30, 1994.

[9] T. Matsui and S. Furui, "A text-independent speaker recognition method robust against utterance variations", *Proc. ICASSP-91*, pp. 377-390, 1991.

[10] M. Savic and J. Sorenson, "Phoneme based speaker verification", *Proc. ICASSP-92*, pp. 165-168, 1992.

[11] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models", *IEEE Trans. Speech and Audio Proc.*, vol. 3, no. 1, pp. 72-83, January 1995.

[12] H. Gish, M. Schmidt and A. Mielke, "A robust, segmental method for text-independent speaker identifcation," *Proc. ICASSP-94*, pp. 145-148, 1994.

[13] J. P. Eatoch and J. S. Mason, "A quantitative assessment of the relative speaker discriminating properties of phonemes," *Proc. ICASSP-94*, pp. 133-136, 1994.

[14] K. T. Assaleh and R. J. Mammone, "Robust cepstral features for speaker identification," *Proc. ICASSP-94*, pp. 129-132, 1994.

[15] C. R. Janowski Jr., T. F. Quatieri and D. A. Reynolds, "Measuring fine structure in speech: Application to speaker identification," in *Proc. ICASSP-95*, pp. 325-328, May 1995.

[16] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," *IEEE Trans. ASSP*, vol. ASSP-29, pp. 254-272, April 1981.

[17] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Proc. Magazine*, pp. 18-32, October 1994.

[18] R. M. Stern, F.-H. Liu, P. J. Moreno and A. Acero, "Signal processing for robust speech recognition", *Proc. ICSLP-94*, vol. 3, pp. 1027-1030, 18-22 September 1994, Yokohama, Japan.

[19] F.-H. Liu, R. M. Stern, A. Acero and P. J. Moreno, "Environment normalization for robust speech recognition using direct cepstral comparison," *Proc. ICASSP-94*, vol. 2, pp. II/61-64, 19-22 April 1994, Adelaide, SA, Australia.

[20] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Ph.D. Thesis, Carnegie Mellon University, 1990.

[21] R. J. Mammone, X. Zhang and R. P. Ramachandran, "Robust speaker recognition", *IEEE Signal Proc. Magazine*, vol. 13, no. 5, pp. 58-71, September 1996.

[22] J.L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains", *IEEE Trans. Speech and Audio Proc.*, vol. 2, no. 2, pp 291-298, April 1994.

[23] C. J. Legetter and P. C. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression", *Proc. Spoken Language Systems Technology Workshop*, pp. 110-115, 1995.

[24] V. Digalakis, D. Rtischev and L. Neumeyer, "Speaker adaptation using constrained reestimation of Gaussian mixtures", *IEEE Trans. Speech and Audio Proc.*, vol. 3, no. 5, pp. 357-366, 1995.

[25] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition", *IEEE Trans. on Speech and Audio Proc.*, pp. 190-202, May 1996.

[26] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," *Proc. ICASSP-94*, vol. 1, pp. 417-420, April 1994.

[27] M. Weintraub and L. Neumeyer, "Constructing telephone acoustic models from a high-quality speech corpus," *Proc. ICASSP-94*, vol. 1, pp. 85-88, April 1994.

[28] S. Lerner and B. Mazor, "Telephone channel normalization for automatic speech recognition," *Proc. ICASSP-92*, pp. 261-264.

[29] A. E. Rosenberg, J. DeLong, B.-H. Juang and F. K. Soong. "The use of cohort normalized scores for speaker verification," *Proc. ICSLP 92*, pp. 599-602, Oct. 12-16, 1992, Banff, Alberta, Canada.

[30] J. J. Godfrey, E. C. Holliman and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development", *Proc ICASSP-92*, pp. I-517-I-520, 1992.

[31] NIST Speaker Recognition Workshop, Johns Hopkins University, Baltimore, Maryland, June 1995.

[32] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition", Prentice Hall, 1993.

[33] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Proc.*, vol. 2, no. 4, pp. 578-589, 1994.

[34] M. Schmidt, H. Gish and A. Mielke, "Covariance estimation methods for channel robust text-independent speaker identification," *Proc. ICASSP-95*, pp.333-336, May 1995.

[35] D. H. Klatt,"A digital filterbank for spectral matching," *Proc. ICASSP-76*, pp. 573-576, April 1976.

[36] B. A. Dautrich, L. R. Rabiner and T. B. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Trans. Acoust. Speech Sig. Proc.*, vol. ASSP-31, no. 4, pp. 793-897, 1983.

[37] D. H. Klatt, "Prediction of perceived phonetic distance from critical-band spectra: A first step," *Proc. ICASSP-82*, pp. 1278-1281, May 1982.

[38] B. Yegnanarayana, "Formant extraction from linear prediction phase spectra," *J. Acoust. Soc. Amer.*, vol. 63, no. 5, pp. 1638-1640, 1978.

[39] H. A. Murthy and B. Yegnanarayana, "Formant extraction from group delay function," *Speech Communication*, vol. 10, pp. 209-221, 1991.

[40] H. A. Murthy and B. Yegnanarayana, "Speech processing using group delay functions," *Signal Proc.*, vol. 22, pp. 259-267, 1991.

[41] B. A. Hanson and H. Wakita, "Spectral slope distance measures with linear prediction analysis for word recognition in noise," *IEEE Trans. Acoust. Speech, Signal Proc.*, vol. ASSP-35, no. 7, pp. 968-973, July 1987.

[42] *Proc. of the LVCSR Workshop*, Oct. 28-30, 1995, Maritime Institute of Technology, Linthicum Heights, Md.

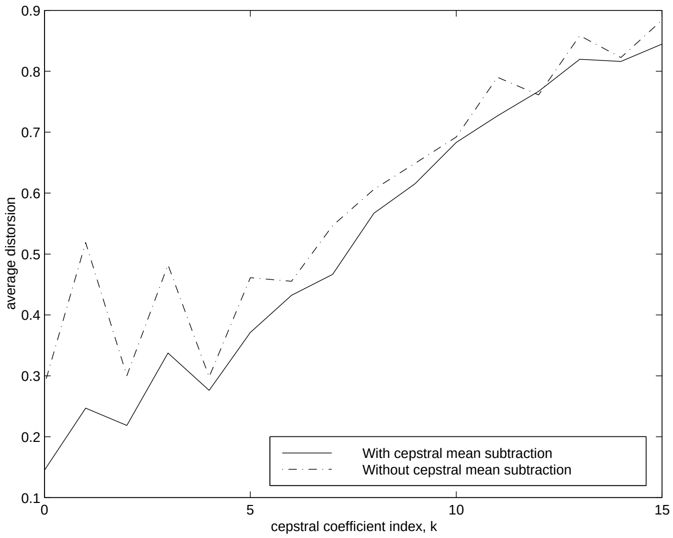

Fig. 1. Average distortion between Sennheiser-recorded cepstral coefficients and their telephone stereo recordings (Stereo-ATIS, 10 sentences per telephone unit and per speaker, 10 telephone units, 13 speakers).

training	testing	CMS	% error
Sennheiser	telephone	yes	16.06
Sennheiser	telephone	no	33.53
Sennheiser	Sennheiser	yes	5.22
Sennheiser	Sennheiser	no	3.85
telephone	telephone	yes	5.81

TABLE I

Speaker identification error rates with matched and mismatched data (Stereo-ATIS, 40-second training, 4-second testing, telephone experiments performed with 10 different handsets).

center-frequency(f_c)	bandwidth(Bw)
(Hz)	(Hz)
$f_c < 1000$	137.5
$1000 \leq f_c \leq 2000$	$1.11\ (f_c)^{0.7}$
$f_c > 2000$	$10.84\ (f_c)^{0.4}$

TABLE II

Bandwidths of the front-end filters as a function of the filter center-frequencies.

N_f	N_c	Min_f	Max_f	B_s	F_w	T_r	% error
22	16	100	3300	N/A	0.6	0.6	33.44

TABLE III

Front-end parameters for the mel-cepstrum feature, before optimization, for the NIST95 database (30-second training, 5-second testing).

I. Experiments on B_s
$N_f = 20, N_c = 17, B_s = \{0.2, 0.4, 0.5, 0.6, 0.8, 0.9, 1.0\}$, $F_w = \{0.0 - 0.5$ in steps of $0.1\}, T_r = \{0.0 - 1.0$ in steps of $0.1\}$, $\text{Min}_f = 300$ Hz, $\text{Max}_f = 3100$ Hz.
II. Experiments on N_c
$N_f = \{20, 24\}, N_c = \{10, 14, 17\}, \{10, 14, 17, 18, 22\}, B_s = 0.8$, $F_w = \{0.0 - 0.2$ in steps of $0.1\}, T_r = \{0.5 - 1.0$ in steps of $0.1\}$, $\text{Min}_f = 300$ Hz, $\text{Max}_f = 3100$ Hz.
III. Experiments on N_f
$N_f = \{20, 24, 28, 32\}, N_c = 17, B_s = 0.8, F_w = \{0.0 - 0.2$ in steps of $0.1\}$, $T_r = \{0.5 - 1.0$ in steps of $0.1\}, \text{Min}_f = 300Hz, \text{Max}_f = 3100Hz$.
IV. Experiments on Min_f and Max_f
$N_f = 24, N_c = 17, B_s = 0.8, F_w = 0.2, T_r = 0.9$ $\text{Min}_f = \{100 - 500Hz\}, \text{Max}_f = \{3000 - 3300Hz\}$.

TABLE IV

Experiments performed on the front-end parameters for the mel-cepstrum feature (NIST95, 30-second training, 5-second testing).

N_f	N_c	Min_f	Max_f	B_s	F_w	T_r	% error
24	17	200	3300	0.8	0.2	0.9	24.89

TABLE V

Best front end for the cepstrum for the NIST95 database (30-second training, 5-second testing).

I. Experiments on B_s
$N_f = 24, B_s = \{0.2, 0.4, 0.6, 0.8, 1.0\}, F_w = \{0.0 - 0.2$ in steps of $0.1\}$, $T_r = \{0.0 - 1.0$ in steps of $0.1\}, \text{Min}_f = 100Hz, \text{Max}_f = 3300Hz$.
II. Experiments on N_f
$N_f = \{20 - 32\}, B_s = \{0.8, 1.0\}, \text{Min}_f = 100Hz, \text{Max}_f = 3300Hz$, $F_w = \{0.0 - 0.2\}, T_r = \{0.0 - 1.0\}$.

TABLE VI

Experiments performed on the front-end parameters for the spectral slope feature (NIST95, 30-second training, 5-second testing).

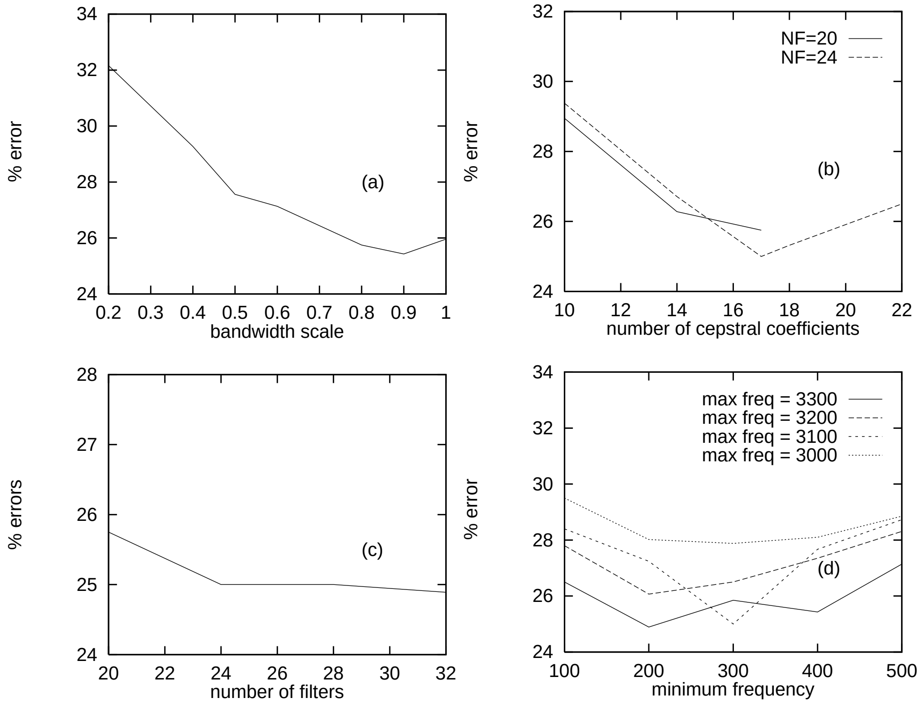

Fig. 2. Optimization of the front-end parameters when varying (a) B_s, (b) N_c, (c) N_f, and (d) Min_f and Max_f (NIST95, 30-second training, 5-second testing).

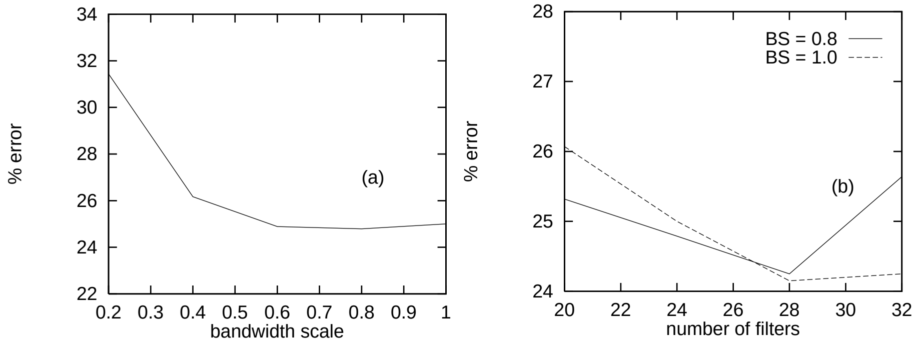

Fig. 3. Optimization of front-end parameters for (a) varying B_s and (b) varying N_f (NIST95, 30-second training, 5-second testing).

N_f	Min_f	Max_f	B_s	F_w	T_r	% error
28	100	3300	1.0	0.1	0.3	24.15

TABLE VII

Best front end for the spectral slope, for the NIST95 database (30-second training, 5-second testing).

Feature	NIST95	SRI − 1	SRI − 2	SRI − 3	SRI − 4	SB45
cepstrum	24.9	26.9	8.2	17.5	17.9	45.11
spectral slope	24.15	27.1	8.28	13.06	16.06	44.81
combined features	23.4	25.4	7.5	13.45	15.94	42.9

TABLE VIII

Percent error rates with the individual and combined features on different databases (NIST95: 30-second training, 5-second testing; SRI-1,2,3,4 1-minute training, 3-second testing (SRI-1: mismatched, 1 handset training, SRI-2: matched, train on all handsets, SRI-3,4: mismatched, train on multiple handsets); SB45: 30-second training, 2-second testing).

Feature	WER in %
cepstrum, non-optimized front-end	75.0
cepstrum, optimized front-end	71.4
spectral slope, optimized front-end	75.0

TABLE IX

Percent word error rates for the development set of the Spanish Callhome LVCSR'95 Evaluation. Different front ends and speech features are compared.

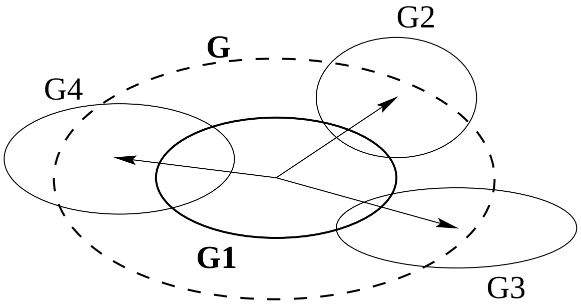

Fig. 4. Clusters of data points in a two-dimensional feature space.

	32 Gaussians		64 Gaussians		128 Gaussians		256 Gaussians	
type of transformation	line 1	line 2	line 1	line 2	line 1	line 2	line 1	line 2
none	26.67	43.10	**25.72**	**42.4**	27.2	43.83	29.56	44.1
LMS affine	23.17	38.22	22.11	36.89	21.28	37.45	**21.22**	**36.72**
translation only	22.89	38.72	22.5	37.61	21.72	38.45	21.95	37.45
scaling only	23.78	40.72	24.72	40.61	25.22	41.61	26.72	42.56

TABLE X

Speaker-ID percent error rate for different model sizes (SRI-digits, 1-minute training on one of two telephone lines, 4-second testing on 10 different lines, 1800 test segments total).

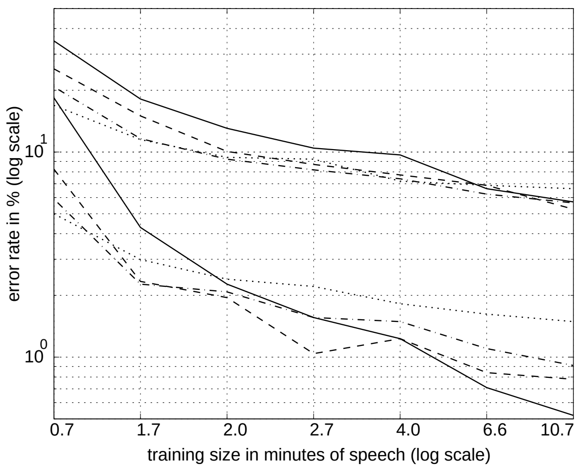

Fig. 5. Speaker-ID error rate as a function of the amount of training data. (Stereo-ATIS, 4-second testing). The upper four curves illustrate the Sennheiser-telephone system performance, the lower four correspond to the matched Sennheiser-Sennheiser system (... = 64 G, -.-. = 128 G, - - = 256 G, — = 512 G).

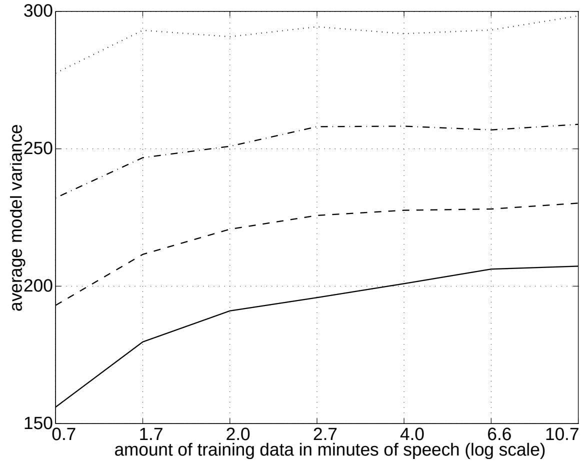

Fig. 6. Average variance along c_2 of four Gaussian speaker models vs. the amount of data used to build the models (... = 64 G, -.-. = 128 G, - - = 256 G, — = 512 G).

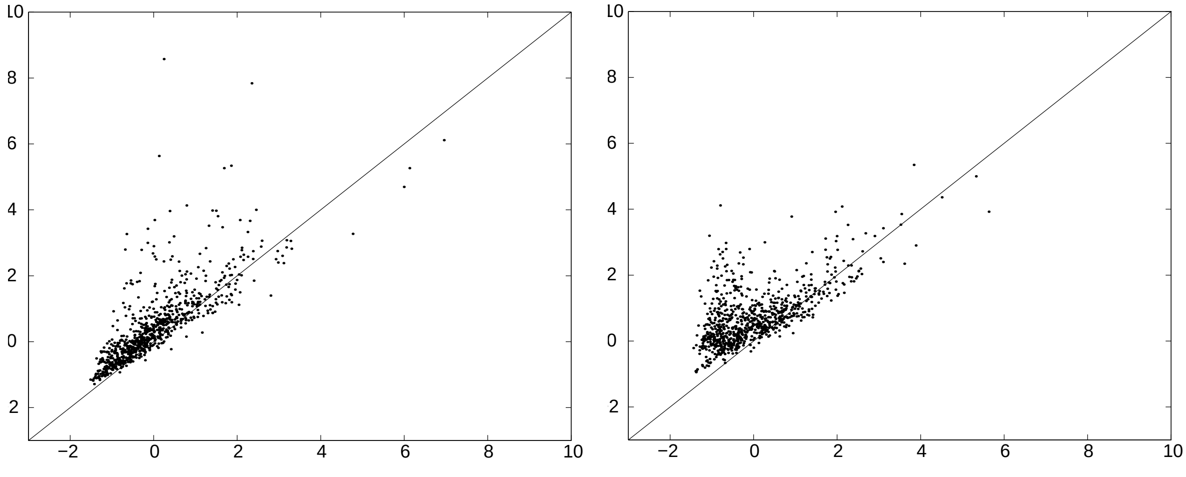

Fig. 7. Pairs of normalized variances, σ_T^2 vs. σ_S^2, along c_1 (left) and c_{17} (right)

training conditions	no transformation			affine transformation			fixed target transf.		
	64 G	128 G	256 G	64 G	128 G	256 G	64 G	128 G	256 G
line 1	**25.72**	27.22	29.56	22.11	21.28	**21.22**	22.00	**21.06**	21.56
line 2	**42.45**	43.83	44.06	36.89	37.45	**36.72**	35.83	35.5	**35.33**
line 3	**42.78**	42.78	45.67	39.28	**37.45**	38.22	38.17	**36.5**	37.61
line 4	52.5	**52.17**	53.28	**46.45**	46.67	47.0	44.72	**43.33**	44.22
line 5	**43.11**	44.17	44.33	41.0	41.45	**40.89**	38.67	**39.72**	41.83
line 6	**43.67**	46.22	48.11	**40.0**	40.67	40.78	38.78	**38.67**	39.89
Average	**41.70**	42.73	44.17	37.62	37.49	**37.47**	36.36	**35.79**	36.74
lines 1 & 2	25.95	**25.56**	32.11	22.28	**20.61**	21.61	21.78	**19.89**	20.67
lines 2 & 3	**31.5**	31.61	39.60	31.0	30.39	**27.83**	30.11	30.22	**28.33**
lines 3 & 4	34.95	**34.72**	41.95	29.56	28.11	**25.00**	27.67	26.39	**23.83**
lines 4 & 5	**31.17**	31.39	36.45	28.39	28.39	**27.61**	27.95	27.72	**27.45**
lines 5 & 6	**28.17**	28.28	35.72	27.5	27.11	**26.27**	**26.72**	26.78	27.33
lines 6 & 1	**20.06**	20.83	28.50	19.28	**18.56**	20.11	18.89	**18.56**	20.33
Average	**28.63**	28.73	35.72	26.33	25.43	**24.74**	25.52	24.93	**24.66**

TABLE XI

Speaker-ID percent error rate for different model sizes, with different variance transformation schemes (SRI-Internal, 1-minute 1-line training or 2-minute 2-line training, 4-second 10-line testing, 1800 test segments total)

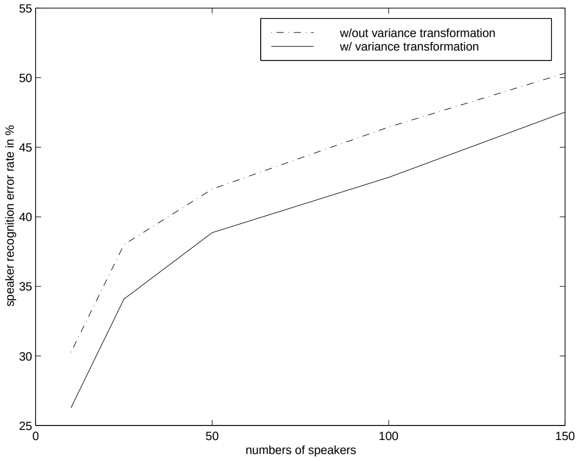

Fig. 8. Speaker-ID error rate as a function of the number of speakers in the database, with and without variance transformation (NIST96, 2-minute training, 30-second testing, mismatched training and testing).

cepstrum	var. transf.	spectral slope	var. transf.	% error
√				24.89
		√		24.15
√	√			23.08
		√	√	22.44
√		√		23.40
√	√	√		22.33
√		√	√	22.54
√	√	√	√	20.83

TABLE XII

Comparison of speaker-ID percent error rate with different systems, on NIST95 (30-second training, 5-second testing, 64 Gaussians per model).

cepstrum	var. transf.	spectral slope	var. transf.	closed-set %error	3% false alarm	10% miss	1% miss
√				24.9	33.8	22.2	52.3
		√		24.1	33.8	20.7	55.3
√	√			23.1	34.3	20.9	47.5
		√	√	22.4	31.0	19.2	48.6
√		√		23.4	32.6	19.6	50.5
√	√	√		22.3	32.4	19.0	44.6
√		√	√	22.5	31.3	18.2	47.6
√	√	√	√	20.8	31.7	16.9	43.5

TABLE XIII

Comparison of open-set speaker recognition error-rate with different systems, on NIST'95 (30-second training, 5-second testing)