Extraction of Speaker Features from Different Stages of DSR Front-ends for Distributed Speaker Verification [*]

Man-Wai Mak and **Chin-Hung Sit**
Center for Multimedia Signal Processing,
Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University,
Hong Kong.

Sun-Yuan Kung
Dept. of Electrical Engineering, Princeton University,
USA.

July 29, 2004

Abstract

The ETSI has recently published a front-end processing standard for distributed speech recognition systems. The key idea of the standard is to extract the spectral features of speech signals at the front-end terminals so that acoustic distortion caused by communication channels can be avoided. This paper investigates the effect of extracting spectral features from different stages of the front-end processing on the performance of distributed speaker verification systems. A technique that combines handset selectors with stochastic feature transformation is also employed in a back-end speaker verification system to reduce the acoustic mismatch between different handsets. Because the feature vectors obtained from the back-end server are vector quantized, the paper proposes two approaches to adding Gaussian noise to the quantized feature vectors for training the Gaussian mixture speaker models. In one approach, the variances of the Gaussian noise are made dependent on the codeword distance. In another approach, the variances are a function of the distance between some unquantized training vectors and their closest code vector. The HTIMIT corpus was

[*] Correspondence should be sent to M.W. Mak, Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. Email: enmwmak@polyu.edu.hk. Tel: (852)27666257. Fax: (852)23628439.

used in the experiments and results based on 150 speakers show that stochastic feature transformation can be added to the back-end server for compensating transducer distortion. It is also found that better verification performance can be achieved when the LMS-based blind equalization in the standard is replaced by stochastic feature transformation.

Keywords: *Distributed speaker verification, DSR, DSR front-end processing, feature transformation.*

1 Introduction

The use of mobile and hand-held devices has become increasingly popular in recent years. While the continuous shrinkage of these devices is a key reason behind their popularity among consumers, the increasing number of tasks that these devices can perform also play an important role. In particular, the ability to surf the Internet and perform financial transactions over the wireless network via the next generation of mobile devices is expected to raise the revenue of the telecommunication industry and the E-commerce market. However, due to their intrinsically small keypads, inputting text and data to these devices is very time consuming and difficult. While speech input is an ideal alternative for this task, mobile phone users tend to use their phones in noisy environment, making robust speech and speaker recognition a challenging task.

Traditionally, speech signals are encoded at the client-side and coded speech signals are transmitted to the server. Recognition is then performed on the server-side after the reconstruction and parameterization of the decoded speech signals. However, it has been found that channel- and codec-distortion can degrade recognition performance significantly (Euler and Zinke, 1994; Lilly and Paliwal, 1996). To address this problem, the European Telecommunications Standard Institute (ETSI) has recently published a front-end processing standard in which feature vectors are extracted at the client-side (Pearce, 2000; ETSI, 2002). The technology is commonly referred to as distributed speech recognition (DSR) in the literature.

Typically, a DSR system can be divided into two parts: a front-end terminal and a back-end server. In the front-end terminal, 13 Mel-frequency cepstral coefficients (from c_0 to c_{12}) and log-energy are extracted from 25ms speech frames at a rate of 100Hz. The features are then compressed using split vector quantization. A 4.8Kbps data channel is established for transmitting the compressed features to a back-end server. In the server-side, the bit-stream is unpacked and recognition can be carried out based on the decompressed feature parameters. Since data in the data channel contain the recognition parameters only, codec distortion can be eliminated. The client/server architecture of DSR systems is highly flexible. In addition to wireless networks, the architecture is also applicable to robust speech recognition over IP networks (Ferandez and Mateo, 2002).

There are studies that compare the recognition performance between the features extracted from the DSR front-end and those extracted from other transcoded speech (Kelleher et al., 2002). The results show that the DSR front-end achieves a significantly better recognition performance. Because mobile phone users typically use their phones in noisy environments, different noise reduction schemes for DSR have been proposed (Andrassy et al., 2001; Noe et al., 2001). These schemes, however, introduce extra computation burden on the front-end processor.

Unlike Andrassy et al. (2001) and Noe et al. (2001) where noise reduction is performed at the client side, this paper proposes and evaluates a distributed speaker recognition system in which channel compensation is performed at the back-end server. Because the LMS algorithm used in the ETSI standard is a kind of linear equalization algorithm, it may not perform satisfactorily on telephone handsets with nonlinear characteristics. To overcome the limitation of the LMS algorithm, this paper incorporates a feature transformation algorithm (Mak and Kung, 2002; Mak et al., 2004; Tsang et al., 2002, 2003) into the back-end recognizer to enhance the robustness of the speaker verification system against handset variations. Experiments with and without using the LMS-based blind equalization were also performed for comparison.

The organization of the paper is as follows. Section 2 provides a brief introduction to speaker recognition and highlights the elements of a speaker verification system. In Section 3, the techniques of stochastic feature transformation and handset identification are briefly discussed. Section 4 introduces two feature-vector perturbation methods that enable us to use quantized feature vectors to train GMM-based speaker models. In Section 5, speaker verification experiments are presented and results are reported in Section 6. Finally, conclusions of the paper are provided in Section 7.

2 Speaker Verification

The goal of automatic speaker recognition is to recognize a speaker from his or her voice (Campbell Jr., 1997; Furui, 1997). Speaker recognition can generally be divided into two categories: *speaker identification* and *speaker verification*. The former determines the identity of an unknown speaker from a group of known speakers, whereas the latter authenticates the identity of a speaker based on his or her own voice. A speaker claiming an identity is called a *claimant*, and an unregistered speaker pretending to be a registered speaker is called an *impostor*. An ideal speaker recognition system should not reject registered speakers (*false rejections*) or accept impostors (*false acceptances*).

Typically, a speaker verification system is composed of a front-end feature extractor, a set of client speaker models, a set of background speaker models, and a decision unit. The feature extractor derives speaker-specific information from the speech signals. It is well known from the source-filter theory of speech production (Fant, 1970) that spectral envelopes implicitly encode vocal-tract shape information (e.g., length and cross-section area) of a speaker and that pitch harmonics encode the glottal source information. Because it is commonly believed that vocal-tract shape varies from speaker to speaker, spectral features, such as linear-predictive cepstral coefficients (LPCCs) (Rabiner and Juang, 1993) and mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980), are often used. A set of speaker models is trained from the spectral features extracted from client utterances. A background model is also trained using the speech of a large number of speakers to represent speaker-independent speech (Reynolds et al., 2000). Basically, the background models are used to normalize the scores of the speaker models to minimize nonspeaker related variability such as acoustic noise and channel effect. To verify a claimant, speaker scores are normalized by the background scores and the resulting normalized score is compared with a decision threshold. The claimant is accepted (rejected) if the score is larger (smaller) than the threshold. As almost perfect verification has become achievable for clean speech, researchers have focused on the problems of transducer mismatches and robustness in recent years.

3 Stochastic Feature Transformation and Handset Identification

Stochastic feature transformation (Sankar and Lee, 1996; Mak and Kung, 2002) is based on the assumption that clean feature vectors $\hat{\mathbf{x}}_t$ can be recovered from distorted vectors $\mathbf{y}_t$ using the transformation

$$\hat{\mathbf{x}}_t = f_\nu(\mathbf{y}_t) = A\mathbf{y}_t + \mathbf{b}, \tag{1}$$

where $\nu = \{A, \mathbf{b}\}$ denotes the transformation parameters and $f_\nu(\cdot)$ is the transformation function. For zeroth-order transformations, A is an identity matrix; for first-order transformations, A is a scaling and rotation matrix. In both cases, $\mathbf{b}$ is a bias vector representing convolutional distortion in the cepstral domain. For computational simplicity, all off-diagonal elements of A are set to zero (i.e., $A = \text{diag}\{a_1, a_2, \ldots, a_D\}$ where D is the feature dimension). The transformation parameters $\nu = \{A, \mathbf{b}\}$ can be estimated by the EM algorithm. More specifically, given the current estimate $\nu' = \{A', \mathbf{b}'\}$ and an M-component Gaussian mixture model (GMM) $\Lambda_X = \{\omega_j, \mu_j, \Sigma_j\}_{j=1}^{M}$ representing clean speech, the new estimate $\nu = \{A, \mathbf{b}\}$ is computed by maximizing an auxiliary function

$$Q(\nu|\nu') = \sum_{t=1}^{T}\sum_{j=1}^{M} h_j(f_{\nu'}(\mathbf{y}_t)) \log \left\{\omega_j p(f_\nu(\mathbf{y}_t)|\mu_j, \Sigma_j, \nu) \left|J_\nu(\mathbf{y}_t)\right|\right\}, \tag{2}$$

where $h_j(f_{\nu'}(\mathbf{y}_t))$ is the posterior probability given by

$$h_j(f_{\nu'}(\mathbf{y}_t)) = P(j|\mathbf{y}_t, \Lambda_X, \nu') = \frac{\omega_j p(f_{\nu'}(\mathbf{y}_t)|\mu_j, \Sigma_j)}{\sum_{l=1}^{M} \omega_l p(f_{\nu'}(\mathbf{y}_t)|\mu_l, \Sigma_l)}, \qquad (3)$$

and $|J_\nu(\mathbf{y}_t)|$ is the determinant of a Jacobian matrix with (r,s)-th entry given by $J_\nu(\mathbf{y}_t)_{rs} = \partial f_\nu(\mathbf{y}_t)_s / \partial y_{t,r}$. For detailed derivation, see (Kung et al., 2004).

In this work, the feature transformation was combined with a handset selector (Tsang et al., 2002; Mak et al., 2004) for robust speaker verification. Specifically, before verification takes place, we compute one set of transformation parameters for each type of handsets that claimants are likely to use. Then, during a verification session, we identify the most likely handset that is used by the claimant and select the best set of transformation parameters accordingly. Let us denote Γ_k as the GMM representing the speech obtained from the k-th handset. Given an utterance, the most likely handset is selected according to

$$k^* = \arg\max_{k=1}^{H} \sum_{t=1}^{T} \log p(\mathbf{y}_t|\Gamma_k), \qquad (4)$$

where H is the number of handsets and $p(\mathbf{y}|\Gamma_k)$ is the density function of the distorted data given the k-th handset. The transformation parameters corresponding to the k^*-th handset are then used for transforming the distorted vectors.

4 Perturbation of Quantized Vectors

Because the feature vectors at the server-side are vector quantized, the distribution of the quantized vectors is discrete. As a result, it is difficult to train a GMM (whose output represents a continuous density function) to fit the quantized data. To overcome this problem, we propose adding zero-mean random vectors to the quantized MFCCs to produce the training vectors. Specifically, the training vectors $\mathbf{u}_t$'s are obtained by

$$\mathbf{u}_t = Q(\mathbf{x}_t) + \boldsymbol{\eta}_t,$$

where $Q(\cdot)$ and $\mathbf{x}_t$ represent the quantization operation and unquantized vectors, respectively, and $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$ are D-dimensional Gaussian vectors with zero mean and diagonal covariance matrix $\Sigma = \text{diag}\{\sigma_1^2, \ldots, \sigma_D^2\}$. Here, we outline two approaches to estimating the values of σ_j's.

4.1 Approach I: Inter-Codeword Distance Dependent Perturbation

In this approach, a VQ codebook $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{S} = \{v_{ij}\}_{j=1}^{D}$, where S is the number of code vectors and D the feature dimension, is derived from the VQ codebooks $\{\mathbf{Q}^{0,1}, \ldots, \mathbf{Q}^{10,11}\}$ defined in the ETSI standard. The code vectors in $\mathcal{V}$ are arranged such that vectors with consecutive indexes are closest to each other in the Euclidean sense. Given the codebook $\mathcal{V}$, the standard derivation of each component of the Gaussian noise $\boldsymbol{\eta}_t$ is found by

$$\sigma_j = \alpha \sqrt{\frac{1}{S-1} \sum_{i=1}^{S-1} \left(v_{ij} - v_{(i+1)j}\right)^2}, \qquad j = 1, \ldots, D, \tag{5}$$

where α is a scaling factor to be found empirically using training data. According to the ETSI standard, $S = 64$ and $D = 12$.

4.2 Approach II: Data-to-Codeword Distance Dependent Perturbation

In approach II, unquantized feature vectors $\{\mathbf{x}_t^{(b_k)}\}_{k=1}^N$ extraced from N background speakers during the training phase are used to build an index table

$$m_{kt} = \arg\min_{i \in \{1,2,\ldots,S\}} \|\mathbf{x}_t^{(b_k)} - \mathbf{v}_i\|, \qquad k = 1, \ldots, N \text{ and } t = 1, \ldots, T_k, \tag{6}$$

where S is the codebook size and T_k is the number of training vectors from the k-th background speaker. Let us define $\boldsymbol{\sigma}'_k \equiv [\sigma'_{k,1}\sigma'_{k,2} \ldots \sigma'_{k,D}]^T$ as the standard deviation vector corresponding to the k-th background speaker. The components of $\boldsymbol{\sigma}'_k$ are found by

$$\sigma'_{k,j} = \sqrt{\frac{1}{T_k} \sum_{t=1}^{T_k} \left(x_{t,j}^{(b_k)} - v_{m_{kt},j}\right)^2}, \quad j = 1, \ldots, D \text{ and } k = 1, \ldots, N, \tag{7}$$

where $x_{t,j}^{(b_k)}$ and $v_{m_{kt},j}$ are the j-th component of $\mathbf{x}_t^{(b_k)}$ and $\mathbf{v}_{m_{kt}}$, respectively. Finally, the diagonal elements of the covariance matrix Σ are calculated by

$$\sigma_j = \frac{\alpha}{N} \sum_{k=1}^{N} \sigma'_{k,j}, \qquad j = 1, \ldots, D, \tag{8}$$

where α is a scaling factor that is estimated using enrollment data. Note that one can access the unquantized feature vectors derived from the background speakers but not from the client speakers. This is because for client speakers, the server can only extract quantized vectors $Q(\mathbf{x}_t)$ from the DSR bit-stream. Because background speakers' speech can be obtained from prerecorded speech copora, it is possible to implement the DSR front-end in software to obtain the unquantized spectral vectors.

Table 1: Feature sets for training and verification. *Unquantized* denotes the MFCCs before vector quantization in the client side, and *quantized* denotes the vector quantized MFCCs extracted from the bit-stream in the server side (see Figure 2).

	Feature Set	
	Training	Verification
Condition A	Unquantized MFCC	Unquantized MFCC
Condition B	Unquantized MFCC	Quantized MFCC
Condition C	Quantized MFCC	Quantized MFCC

4.3 Comparing Approach I and Approach II

Figure 1 shows the projection of the quantized feature vectors on the c_1-c_2 plane before and after random vectors were added using Approach I and Approach II. We can observe from the figure that both approaches can make the distribution of the perturbed features very similar to the unquantized one. As a result, training the GMM speaker models will become possible.

One should bear in mind that the objective of adding the random vectors is to enable the EM algorithm to train the GMM speaker models instead of recovering the unquantized feature vectors from their quantized counterparts. In this respect, the noise variance σ_j^2 should not be too large; otherwise the speaker characteristics contained in the noise-added vectors cannot be maintained. In fact, in a preliminary experiment, we have tried removing the square root in Eq. (5) and found that the resulting speaker models cannot discriminate client speakers from impostors.

5 Experiments

5.1 Speech Data

The HTIMIT corpus (Reynolds, 1997) was used to evaluate the speaker features extracted from different stages of the DSR front-end. HTIMIT was obtained by playing a subset of the TIMIT corpus (Fisher et al., 1986) through four carbon button telephone handsets (cb1–cb4), four electret handsets (el1–el4), a portable handset (pt1), and a Sennheizer head-mounted microphone (senh). Unlike other telephone speech databases where no handset labels are given, every utterance in HTIMIT is labelled with a handset name (cb1–cb4, el1–el4, pt1 or senh). This feature makes HTIMIT appropriate for the study of transducer effect in speech/speaker recognition, and HTIMIT has been used in speaker recognition studies (Quatieri et al., 2000).

Speakers in the corpus were divided into a customer set (50 male and 50 female) and an impostor set (25 male and 25 female). Each speaker in the corpus spoke two dialectal sentences (the SA sentence set), five phonetically-compact sentences (the SX sentence set) and three phonetically-diverse sentences (the SI sentence set). Each speaker spoke the same set of sentences in the SA sentence set. In the SX sentence set, some speakers spoke the same sentences. However, all sentences in the SI sentence set are different. Therefore, the HTIMIT corpus allows us to perform text-independent speaker verification by using SA and SX sentence sets as the training set and the SI sentence set as the test set.

5.2 Quantized and Unquantized Feature Vectors

According to the ETSI standard for DSR systems (ETSI, 2002), blind equalization is applied to the spectral features (MFCCs) during the feature extraction stage. To compare the performance of stochastic feature transformation against blind equalization, two sets of experiments were performed. In one set of experiments, LMS-based blind equalization was applied, while in another set, no blind equalization was applied. Two sets of feature vectors, an *unquantized* set and a *quantized* set, were extracted from different stages of the front-end processor of the ETSI standard (cf. Figure 2). More precisely, the unquantized set was extracted before feature compression in the terminal front-end while the quantized set was extracted just after server feature processing. To obtain a better comparison, speaker verification was performed under three conditions shown in Table 1.

Although the ETSI standard specifies that a total of 39 coefficients ($c_1, c_2, \ldots, c_{12}, \{\ln E \;\&\; c_0\}$, and their first- and second-derivatives) per speech frame can be extracted from server-end bit-streams, only 12 MFCCs ($c_1, c_2, \ldots, c_{12}$) per frame were used in the experiments reported here. Therefore, the values of S and D in Eqs. (5) through (8) were set to 64 and 12, respectively.

5.3 Enrollment Procedures

Similar to our previous work (Mak and Kung, 2002; Mak et al., 2004; Yiu et al., 2003; Tsang et al., 2002), we trained a personalized 32-center GMM to model the characteristics of each client speaker in the system.[1] The feature vectors derived from the SA and SX sentence sets of the corresponding speaker were used for training, i.e., 7 sentences per GMM. A collection of all SA and SX sentences uttered by all speakers in the customer set were used to train a 64-center

[1] We chose not to use MAP adaptation (Reynolds et al., 2000) because we have enough data to create individual speaker models. Our recent study (Sit et al., 2004) also shows that when sufficient training data are available, the maximum likelihood approach can create better speaker models than the MAP approach.

Table 2: Equal error rates (EERs) for different values of α using (a) Approach I and (b) Approach II. Speech data from handset cb1 were used and Condition C was used for training and verification.

α	0.250	0.280	0.300
EER	7.31%	6.63%	7.00%

(a) Approach I

α	0.139	0.140	0.143
EER	6.76%	5.85%	7.31%

(b) Approach II

GMM background model (Reynolds et al., 2000). The handset senh was used as the enrollment handset and its utterances were considered to be clean.

The optimum values of α in Eqs. (5) and (8) were determined empirically using the speech data of handset cb1. Table 2 shows the equal error rates—the rate at which the probability of false acceptance is equal to that of false rejection—for different values of α. Based on Table 2, we set $\alpha = 0.28$ for Approach I and $\alpha = 0.140$ for Approach II in all experiments.

5.4 Feature Transformation

The clean utterances (from handset senh) of 10 speakers were used to create a 2-center GMM clean model Λ_X, i.e., $M = 2$ in Eq. (2). Using this model and the estimation algorithms described in Section 3, a set of feature transformation parameters ν were computed for each handset. In particular, the utterances from handset senh were considered as clean and were used to create the clean speech model, while those from other 9 handsets (cb1–cb4, el1–el4, and pt1) were used as distorted speech. Zeroth-order ($f_\nu(\mathbf{y}_t) = \mathbf{y}_t + \mathbf{b}$) and first-order ($f_\nu(\mathbf{y}_t) = A\mathbf{y}_t + \mathbf{b}$) transformations were used in this work.

5.5 Verification Procedures

During verification, a vector sequence $\mathbf{Y}$ derived from a claimant's utterance (SI sentence) was fed to a handset selector. According to the outputs of the handset selector, a set of transformation parameters was selected. The features were transformed and then fed to a 32-center GMM speaker model ($\mathcal{M}_s$) and the 64-center GMM background model ($\mathcal{M}_b$) to obtain a normalized score:

$$s(\mathbf{Y}) = \log p(\mathbf{Y}|\mathcal{M}_s) - \log p(\mathbf{Y}|\mathcal{M}_b). \tag{9}$$

The normalized score $s(\mathbf{Y})$ was compared with a threshold for decision making. For ease of comparison, we collect the scores of 100 client speakers, each being impersonated by 50 impostors, to compute the speaker-independent equal error rate (EER) and to produce a detection error trade-off curve (Martin et al., 1997). Therefore, speaker-independent decision thresholds were used, and for each handset in an experimental setting, there were 300 client speaker trials (100 client speakers × 3 sentences per speaker) and 15,000 impostor trials (50 impostors per client speaker × 100 client speakers × 3 sentences per impostor).

Table 3: EERs (in %) with LMS blind equalization for matched and mismatched handsets in three training and verification conditions. ST0 and ST1 stand for zeroth- and first-order stochastic transformations, respectively. C(I) and C(II) represent Condition C in Table 1 with Approach I and Approach II being used for creating the training vectors, respectively.

	Trans.	**Equal Error Rate (%)**										
Cond.	**Method**	cb1	cb2	cb3	cb4	el1	el2	el3	el4	pt1	Average	senh
A	Baseline	7.35	6.54	15.96	12.34	5.68	12.63	11.74	12.26	9.51	10.45	3.54
A	ST0	6.35	5.82	15.30	10.77	5.46	11.13	12.17	9.19	8.26	9.38	3.52
A	ST1	6.19	5.57	14.53	10.89	5.34	10.90	11.65	8.86	8.57	9.17	3.66
B	Baseline	7.55	7.02	15.92	13.01	5.99	12.14	11.98	11.26	9.66	10.50	2.97
B	ST0	5.95	5.87	16.19	12.22	5.81	10.98	12.92	9.40	9.15	9.83	2.97
B	ST1	5.92	5.58	14.88	11.29	5.88	10.63	11.50	9.02	8.91	9.29	2.98
C (I)	Baseline	6.63	6.55	16.59	11.67	5.33	10.76	10.71	9.97	8.83	9.67	3.15
C (I)	ST0	6.60	5.65	15.97	11.58	6.07	10.07	11.89	9.57	8.66	9.57	3.26
C (I)	ST1	6.60	5.24	15.08	11.81	5.53	10.19	11.66	8.80	8.34	9.25	3.22
C (II)	Baseline	6.56	6.29	15.51	10.87	5.22	10.92	10.28	10.65	8.69	9.44	2.78
C (II)	ST0	5.85	5.62	15.67	10.85	5.10	9.43	10.30	7.86	7.97	8.74	2.94
C (II)	ST1	6.32	5.47	14.18	10.24	5.54	9.24	9.90	7.43	8.06	8.49	2.98

6 Results and Discussions

Tables 3 and 4 show the EERs achieved by the feature transformation approach and the baseline (without feature transformation) under different training and verification conditions for the cases with and without LMS blind equalization, respectively. All error rates are based on the scores of 100 genuine speakers and 50 impostors. The average EERs under the label *Average* were computed by taking the average of all the EERs corresponding to the nine mismatched handsets (cb1–cb4, el1–el4, and pt1). Likewise, the EERs under *senh* correspond to the EERs obtained by using the enrollment handset (senh) for verification.

Table 4: EERs (in %) without LMS blind equalization for matched and mismatched handsets in three training and verification conditions. ST0 and ST1 stand for zeroth- and first-order stochastic transformations, respectively. C(I) and C(II) represent Condition C in Table 1 with Approach I and Approach II being used for creating the training vectors.

	Trans.	**Equal Error Rate (%)**										
Cond.	**Method**	cb1	cb2	cb3	cb4	el1	el2	el3	el4	pt1	Average	senh
A	Baseline	11.10	12.95	26.40	22.86	7.54	23.01	13.67	21.60	18.20	17.48	2.19
A	ST0	5.27	3.64	13.39	8.30	4.45	8.01	10.68	6.01	5.51	7.25	2.19
A	ST1	5.28	3.58	13.29	8.57	4.45	7.89	9.88	5.56	5.78	7.14	2.20
B	Baseline	11.15	13.58	26.14	23.36	7.35	19.51	13.66	21.54	18.27	17.17	2.18
B	ST0	4.88	3.91	15.38	8.96	5.88	11.84	10.65	6.55	7.77	8.42	2.24
B	ST1	4.88	3.88	13.94	8.21	5.91	11.02	10.86	5.57	8.34	8.07	2.24
C (I)	Baseline	10.52	13.46	24.51	20.65	6.44	18.05	11.90	17.92	17.82	15.70	2.45
C (I)	ST0	4.41	3.33	13.23	7.65	4.42	7.43	8.91	5.59	6.60	6.84	2.49
C (I)	ST1	4.50	3.25	11.55	7.25	4.37	7.15	8.94	5.46	5.87	6.48	2.52
C (II)	Baseline	10.32	13.52	24.22	19.61	6.42	18.76	11.89	19.35	17.00	15.68	2.26
C (II)	ST0	4.43	3.61	12.65	7.98	4.01	8.30	9.33	5.52	6.25	6.90	2.27
C (II)	ST1	4.30	3.84	11.66	7.78	4.01	7.86	8.92	5.55	6.60	6.72	2.28

6.1 Performance Under Different Training and Verification Conditions

Comparing the results in the handset-mismatch cases of Conditions A and B in Table 3 and Table 4, it is obvious that the error rates are generally higher in Condition B regardless of whether LMS blind equalization is used or not. This is because in Condition B, unquantized MFCCs were used for training whereas quantized MFCCs were used for verification. As a result, in addition to handset mismatches between enrollment and verification, there are also feature mismatches.

6.2 Performance Under Handset-Match and Handset-Mismatch Conditions

To facilitate discussion and comparison of results, we extract the figures in Tables 3 and 4 and summarize them in Figures 3 and 4. In particular, Figure 3 depicts the effect of LMS blind equalization on handset-match and handset-mismatch situations in the baseline case, and Figure 4 compares the average EERs for matched and mismatched handsets using stochastic transformation under the three training and verification conditions.

The results in Figure 3 show that LMS blind equalization can reduce error rates under handset-mismatch conditions, but it degrades the performance when enrollment and verification use the same handset. From Figure 4(b), we can observe that for mismatched handsets, the amount of error reduction achieved by stochastic feature transformation is larger than that by LMS. However, we can observe from Figure 4(a) that, for the handset-match conditions, feature transformation is inferior to the baseline. This is mainly because in the cases where both training and verification use the same handset (senh), the handset selectors may misclassify some of the senh utterances as recorded from other handsets. As a result, some of the LMS-feature vectors may not be transformed correctly. Although this handset misclassification leads to performance degradation in handset-match conditions, the degradation is insignificant when compared to the performance improvement in the handset-mismatch cases. The combination of handset selector and feature transformation can therefore still help improve the performance when the handset-mismatch conditions are also taken into account.

6.3 Effect of LMS Blind Equalization on Verification Performance

Evidently, Figure 4(b) shows that stochastic feature transformation gives the largest error reduction in handset-mismatch conditions when both training and verification vectors are extracted from the bit-streams at the server side (Condition C). With LMS blind equalization, feature transformation reduces the average EER by 4.35% for Approach I (from 9.67% to 9.25%). The percentage reduction increases to 58.70% for Approach I (from 15.70% to 6.48%) when LMS was not used. Similarly, with LMS blind equalization, the average percentage reduction in EER is about 10.13% for Approach II (from 9.44% to 8.49%) and is about 57.10% (from 15.68% to 6.72%) for Approach II without using LMS.

Figure 5 shows the detection error tradeoff curves of handset cb1 under Condition C. The curves allow us to observe the effect of LMS blind equalization on speaker verification performance under various speaker-independent decision thresholds. Three observations can be obtained from the figure. First, the figure shows that feature transformation performs better than the baseline system. Second, LMS blind equalization improves the performance of the baseline system (compare Curves I and II). Third, the transformation technique is more effective when blind equalization is not applied (compare Curves III and IV). This suggests that using stochastic feature transformation alone is even better than using stochastic feature transformation with LMS.

While the results suggests that stochastic feature transformation can improve the verification performance of DSR systems, the performance will be further improved if LMS blind equalization is not applied. This is reasonable because LMS blind equalization on one hand minimizes convolutional distortion, but on the other it removes some of the speaker characteristics. While both LMS and stochastic feature transformation shift the positions of the feature distributions in the feature space, LMS will shift the means of the speaker distributions towards the origin of the feature space. This may reduce the discriminative power of the speaker and background models.

7 Conclusions

Features extracted from different stages of the ETSI-DSR front-end in the context of distributed speaker verification have been evaluated. It was found that the best performance can be obtained from the condition where feature vectors for both training and verification are extracted from the server side. Also, the technique of combining stochastic feature transformation and handset identification has been applied to the extracted features. Results show that the transformation technique can significantly reduce the error rates of an ETSI-compliance distributed speaker verification system. The recognition accuracy is even higher when LMS blind equalization in the ETSI standard is replaced by the feature transformation approach.

8 Acknowledgment

The authors would like to thank the reviewers for their valuable suggestions and comments, which have improved the quality of this paper. This work was in part supported by the The Hong Kong Polytechnic University, Grant No. A-PE44 and Research Grant Council of the Hong Kong SAR (Project No. CUHK 1/02C).

References

Andrassy, B., Hilger, F., and Beaugeant, C. (2001). Investigations on the combination of four algorithms to increase the noise robustness of a DSR front-end for real world car data. In: *Proc. IEEE Workshop on Automatic Speech Recognition 2001, ASRU'01*, pp. 115–118.

Campbell Jr., J. P. (1997). Speaker Recognition: A Tutorial. *Proc. IEEE*, **85**(9), 1437–1462.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on ASSP*, **28**(4), 357–366.

ETSI (2002). Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms. Technical Report ETSI ES 202 050 V1.1.1 (2002-10), European Telecommunications Standard Institute.

Euler, S. and Zinke, J. (1994). The influence of speech coding algorithms on automatic speech recognition. In: *Proc. ICASSP'94*, pp. 621–624.

Fant, G. (1970). *Title Acoustic Theory of Speech Production with Calculations Based on X-Ray Studies of Russian Articulations.* Mouton: The Hague, 2nd edition.

Docio-Fernandez, L. and Garcia-Mateo, C. (2002). Distributed speech recognition over IP networks on the AURORA 3 database. In: *Proc. ICSLP'02*, pp. 461–464.

Fisher, W. M., Doddington, G. R., and Goudie-Marshall, K. M. (1986), The DARPA speech recognition research database: Specifications and status. In: *Proc. DARPA Workshop on Speech Recognition*, pp. 93–99.

Furui, S. (1997). Recent advances in speaker recognition. *Pattern Recognition Letters*, **18**, 859–872.

Kelleher, H., Pearce, D., Ealey, D., and Mauuary, L. (2002). Speech recognition performance comparison between DSR and AMR transcoded speech. In: *Proc. ICSLP'02*, pp. 1873–1876.

Kung, S. Y., Mak, M. W., and Lin, S. H. (2004). *Biometric Authentication: A Machine Learning Approach.* New Jersey: Prentice Hall.

Lilly, B. T. and Paliwal, K. K. (1996). Effect of speech coders on speech recognition performance. In: *Proc. ICSLP*, vol. 4, pp. 2344–2347.

Mak, M. W. and Kung, S. Y. (2002). Combining stochastic feature transformation and handset identification for telephone-based speaker verification. In: *Proc. ICASSP'02*, Vol. 1, pp. 701–704.

Mak, M. W., Tsang, C. L., and Kung, S. Y. (2004). Stochastic feature transformation with divergence-based out-of-handset rejection for robust speaker verification. *EURASIP J. on Applied Signal Processing*, **4**, 452–465.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In: *Proc. Eurospeech'97*, pp. 1895–1898.

Noe, B., Sienel, J., Jouvet, D., Mauuary, L., Boves, L., de Veth, J., and de Wet, F. (2001). Noise reduction for noise robust feature extraction for distributed speech recognition. In: *Proc. Eurospeech 2001*, Vol. 1, pp. 433–436.

Pearce, D. (2000). Enabling new speech driven services for mobile devices: an overview of the ETSI standards activities for distributed speech recognition front-ends. In: *Proc. AVIOS 2000: The Speech Application Conference.*

Quatieri, T. F., Reynolds, D. A., and O'Leary, G. C. (2000). Estimation of handset nonlinearity with application to speaker recognition. *IEEE Trans. on Speech and Audio Processing*, **8**(5), 567–584.

Rabiner, L. and Juang, B. H. (1993). *Fundamentals of Speech Recognition.* New Jersey: Prentice-Hall.

Reynolds, D. A. (1997). HTIMIT and LLHDB: speech corpora for the study of handset transducer effects. In: *Proc. ICASSP'97*, Vol. 2, pp. 1535–1538.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using Adapted Gaussian mixture models. *Digital Signal Processing*, **10**, 19–41.

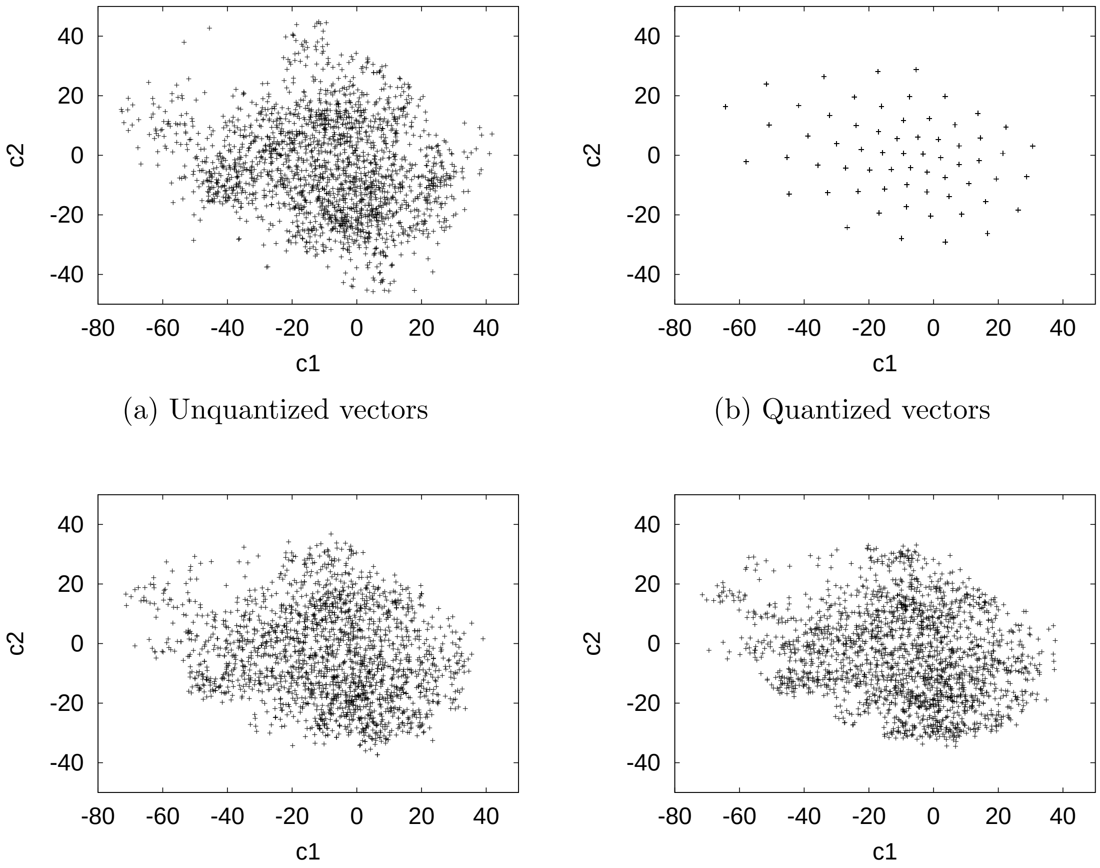
Sankar, A. and Lee, C. H. (1996). A maximum-likelihood approach to stochastic matching for robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, **4**(3), 190–202.

Sit, C. H., Mak, M. W., and Kung, S. Y. (2004). Maximum likelihood and maximum a posteriori adaptation for distributed speaker recognition systems. In: *Proc. International Conference on Biometric Authentication (ICBA'04).* Hong Kong, pp. 640–647, Springer.

Tsang, C. L., Mak, M. W., and Kung, S. Y. (2002). Divergence-based out-of-class rejection for telephone handset identification. In: *Proc. ICSLP'02*, pp. 2329–2332.

Tsang, C. L., Mak, M. W., and Kung, S. Y. (2003). Cluster-dependent feature transformation for telephone-based speaker verification. In: *Proc. International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'03)*, Surrey, U.K., pp. 86–94.

Yiu, K. K., Mak, M. W., and Kung, S. Y. (2003). Environment adaptation for robust speaker verification. In: *Proc. Eurospeech'03*, Geneva, pp. 2973–2976.



(a) Unquantized vectors

(b) Quantized vectors

(c) Perturbed vectors (Approach I)

(d) Perturbed vectors (Approach II)

Figure 1: Projection of training vectors on the c_1-c_2 space. (a) Unquantized feature vectors. (b) Quantized feature vectors before adding random vectors. (c) Quantized training vectors after adding random vectors to the quantized vectors in (b) using Approach I (Inter-codeword distance dependent perturbation). (d) Similar to (c) but use Approach II (Data-to-codeword distance dependent perturbation).

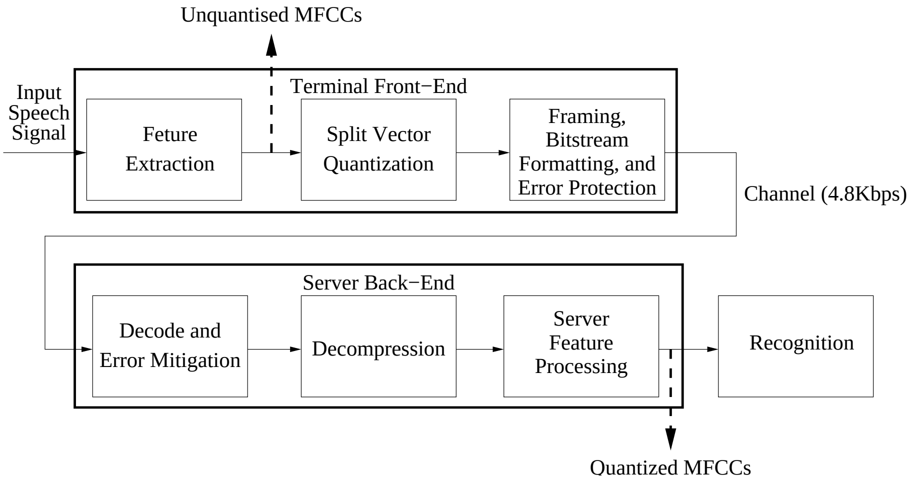


Figure 2: Feature extraction and processing in an ETSI-compliance DSR system.

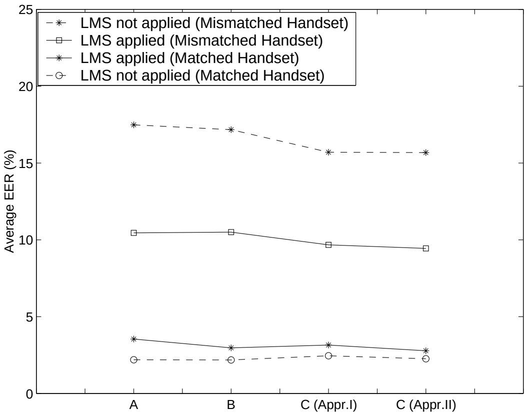
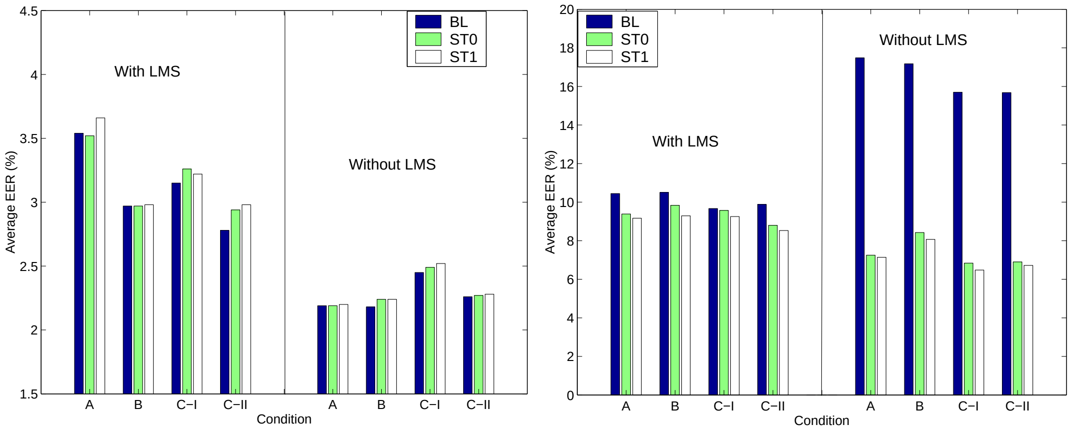


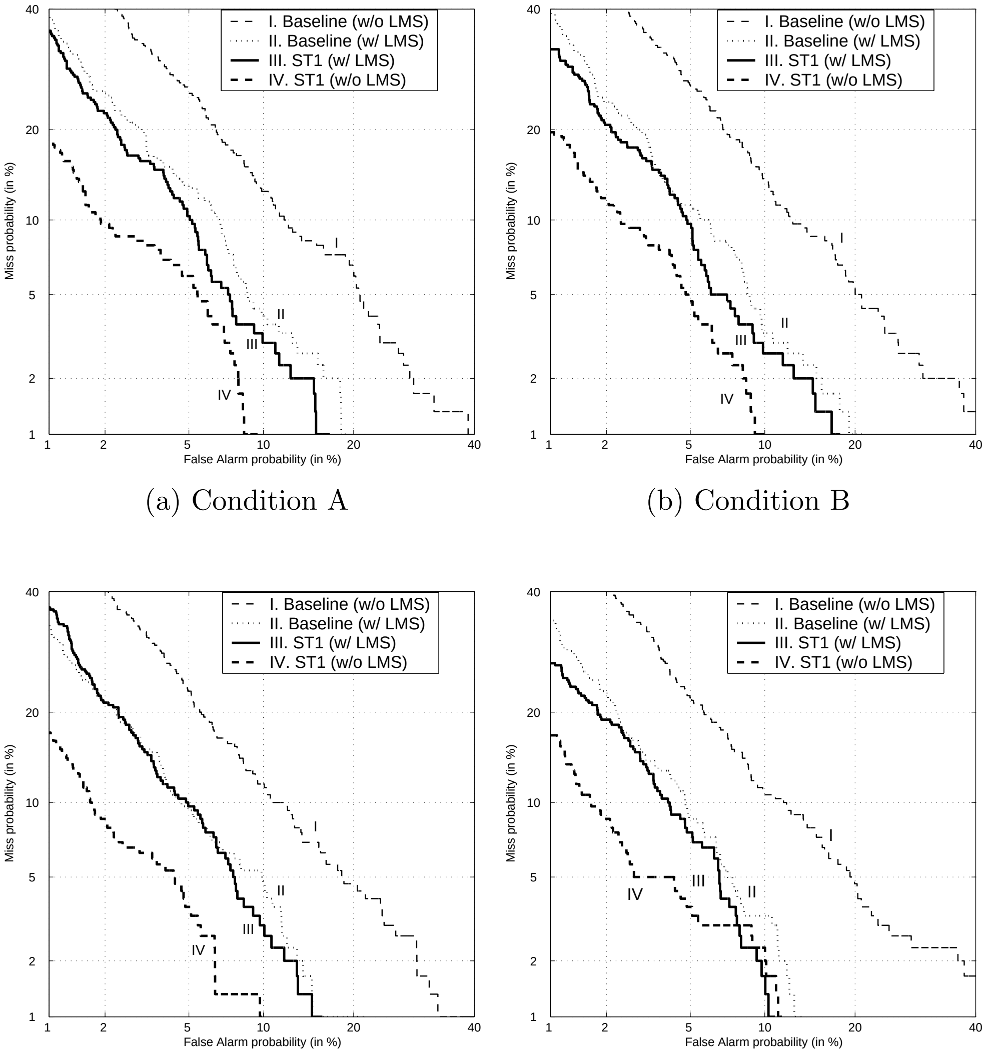
Figure 3: Average EERs demonstrating the effect of LMS blind equalization on both handset-match and handset-mismatch conditions in the baseline case.



(a) Matched Handset

(b) Mismatched Handsets

Figure 4: Summary of the average EERs of (a) matched handsets and (b) mismatched handsets in Tables 3 and 4. BL, ST0, and ST1 stand for baseline, zeroth-order stochastic transformation, and first-order stochastic transformation, respectively.



(a) Condition A

(b) Condition B

(c) Condition C with Approach I

(d) Condition C with Approach II

Figure 5: DET curves for handset cb1 under (a) Condition A (b) Condition B (c) Condition C using Approach I and (d) Condition C using Approach II. ST1 stands for first-order stochastic feature transformation.