B.L. Pellom, J.H.L. Hansen, "An Experimental Study of Speaker Verification Sensitivity to Computer Voice-Altered Imposters," IEEE ICASSP-99: Inter. Conf. on Acoustics, Speech, and Signal Processing, vol. 2, pp. 837-840, Phoenix, Arizona, March 1999.

An Experimental Study of Speaker Verification Sensitivity to Computer Voice-Altered Imposters

Bryan Pellom, John H.L. Hansen

Robust Speech Processing Laboratory
Center for Spoken Language Research
University of Colorado Boulder, Campus Box 594
(Express Mail: 3215 Marine Street, Room E-265)
Boulder, Colorado 80309-0594
303 – 735 –5148 (Phone) 303 – 735 – 5072 (Fax)
http://cslr.colorado.edu/
John.Hansen@colorado.edu, (email)

IEEE ICASSP-99: Inter. Conf. On Acoustics, Speech, and Signal Processing, Phoenix, Arizona, March 1999.

AN EXPERIMENTAL STUDY OF SPEAKER VERIFICATION SENSITIVITY TO COMPUTER VOICE–ALTERED IMPOSTERS

Bryan L. Pellom and John H.L. Hansen

Robust Speech Processing Laboratory
Duke University, Box 90291, Durham, NC 27708-0291
http://www.ee.duke.edu/Research/Speech bp@ee.duke.edu jhlh@ee.duke.edu

ABSTRACT

This paper investigates the relative sensitivity of a GMM–based voice verification algorithm to computer voice–altered imposters. First, a new trainable speech synthesis algorithm based on trajectory models of the speech Line Spectral Frequency (LSF) parameters is presented in order to model the spectral characteristics of a target voice. A GMM–based speaker verifier is then constructed for the 138 speaker YOHO database and shown to have an initial equal–error rate (EER) of 1.45% for the case of casual imposter attempts using a single combination–lock phrase test. Next, imposter voices are automatically altered using the synthesis algorithm to mimic the customer's voice. After voice transformation, the false acceptance rate is shown to increase from 1.45% to over 86% if the baseline EER threshold is left unmodified. Furthermore, at a customer false rejection rate of 25%, the false acceptance rate for the voice–altered imposter remains as high as 34.6%.

1. INTRODUCTION

There has been considerable interest in voice verification technology over the past twenty–five years. Much attention has been devoted to methods for better characterization of the customer voice or at improving the background model for the imposter (e.g., [1, 2]). Earlier studies by Rosenberg and Sambur [3], for example, investigated the sensitivity of voice verification algorithms to professional trained human mimics. That study found that the sensitivity to human impersonation is relatively low. For example, the false acceptance rate was shown to increase from 1% (for casual imposters) to only 4% (for professional mimics). In recent years, however, several algorithms have been proposed for computer–aided voice conversion (e.g., [4, 5]). While voice conversion approaches continue to mature, it is worth establishing the current sensitivity of voice verification systems to attack by computer altered imposter voices. Therefore, this study first presents a new algorithm for speaker–dependent trainable speech synthesis and subsequently evaluates the approach by utilizing the algorithm as a pre–processor for the imposter voice prior to voice verification.

2. TRAINABLE SPEECH SYNTHESIS: LSF TRAJECTORY MODELING

Voice conversion techniques attempt to learn a functional mapping between an input source voice and desired target voice. Trainable synthesis algorithms, on the other hand, model the target voice directly. Consequently, algorithms of this type are also useful for text–to–speech synthesis applications. The proposed trainable synthesis algorithm (referred to as LSF–STM) is based on an extension of the Stochastic Trajectory Model (STM) approach proposed previously by Gong and Haton for continuous speech recognition [6, 7]. Highlights of the LSF–STM algorithm follow, and a detailed description of the modeling strategy can be found in [8].

This work work was supported in part by a National Science Foundation Graduate Research Fellowship.

2.1. Speech Analysis Method

The analysis algorithm partitions speech into monophone units and extracts trajectory representations of the observed feature sequences. Specifically, speech from the target voice is first partitioned at the phoneme level by an automatic HMM–based time–alignment procedure [9]. Using a pitch tracking algorithm, the pre–emphasized input speech is analyzed pitch–synchronously during voiced excitation and at a constant rate during unvoiced excitation [10]. At each analysis instant, a short–time windowed waveform is extracted by applying a Hanning window centered around the epoch location. A Pth order LP analysis is then performed ($P = 10$ for 8 kHz speech). The LPC vector is transformed into a P–dimensional Line Spectral Frequency (LSF) vector [11]. Next, in order to characterize the temporal velocity of the spectral parameters, delta–LSF (ΔLSF) parameters are computed by a linear regression of the two adjacent LSF vectors surrounding the current analysis instant. The P–dimensional ΔLSF vector is appended onto the static LSF vector resulting in an observation containing $2P$ elements.

An illustration of the feature extraction and trajectory encoding process is shown in Fig. 1. Here, the time–axis of the pitch–synchronous parameter sequence (Fig. 1B) is resampled to Q uniformly spaced points resulting in a final Q–state trajectory representation (Fig. 1C). In other words, an observation sequence of L frames, $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L\}$ is mapped to a Q–state trajectory, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q\}$. The qth state of the trajectory contains a vector of LSF/ΔLSFs,

$$\mathbf{x}_q = [\omega_1(q), \ldots, \omega_P(q), \Delta\omega_1(q), \ldots, \Delta\omega_P(q)] \qquad (1)$$

where the jth element of $\mathbf{x}_q$ is represented as $\mathbf{x}_q[j]$ and $\omega_j(q)$ represents the jth LSF.

Additional information is extracted for each trajectory unit to aid in model estimation. For example, the *acoustic class* of the left and right adjacent phones is used to encode the phonetic context of each training pattern [12]. Vowels are additionally distinguished in the training data by 3 lexical stress markers (primary, secondary, or no stress).

2.2. Synthesis Trajectory Model

The synthesis model consists of a Q–state trajectory representation. The state count is assumed fixed for all phones ($Q = 5$) and the number of modeled trajectories for each phoneme is based on the amount of available training data. Each monophone synthesis model is comprised of a set of K

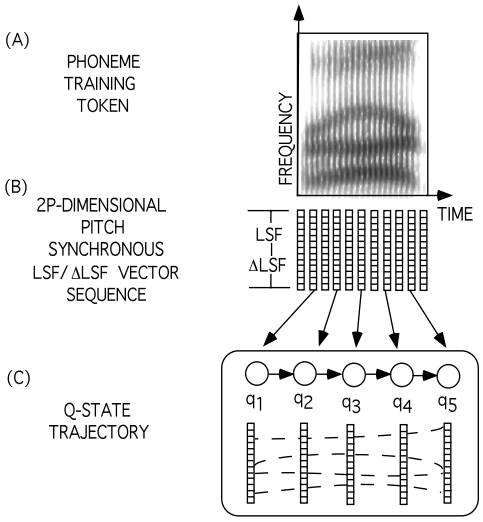

Figure 1: Illustration of trajectory characterization. In (A) the wideband spectrogram of a hypothetical training pattern is shown. In (B) LSF and ΔLSF vectors are pitch–synchronously extracted from the waveform. Finally, in (C) the LSF and ΔLSF vectors are resampled to form a Q–state trajectory.

LSF/ΔLSF spectral trajectories where the kth trajectory, $\mathbf{T}_k$, is described by,

1. Q mean LSF/ΔLSF vectors, $(\mu_{qk}; q = 1, \ldots, Q)$,
2. Q covariance matrices (Σ_{qk}). The covariance matrices are assumed to be diagonal and thus characterized by a set of Q state–dependent LSF/ΔLSF variance vectors, $(\sigma^2_{qk}; q = 1, \ldots, Q)$,
3. The probability of generating an observation from the kth modeled trajectory, $p_k = \mathrm{P}(\mathbf{T}_k)$. The probabilities follow the sum–to–one constraint, $\sum_{k=1}^{K} p_k = 1$.

Unlike the recognition STM formulated by Gong and Haton, the synthesis STM computes a perceptually motivated distance between an observed training data trajectory and a modeled trajectory. Specifically, the distance between the observed trajectory $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_Q\}$ and kth modeled mean trajectory $\mathbf{T}_k = \{\mu_{1k}, \ldots, \mu_{Qk}\}$ is given by,

$$\mathrm{d}(\mathbf{X}, \mathbf{T}_k) = \alpha \sum_{q=1}^{Q} \sum_{j=1}^{P} \mathbf{c}_q[j] \left(\mathbf{x}_q[j] - \mu_{qk}[j]\right)^2 + (1\text{–}\alpha) \sum_{q=1}^{Q} \sum_{j=P+1}^{2P} \mathbf{c}_q[j-P] \left(\mathbf{x}_q[j] - \mu_{qk}[j]\right)^2 . \quad (2)$$

Here, α $(0 \leq \alpha \leq 1)$ is used to adjust the relative contributions of the static $(j = 1, 2, \ldots, P)$ and dynamic $(j = P+1, \ldots, 2P)$ LSF parameters in the distortion function. The jth LSF parameter weighting term for the qth modeled state, $\mathbf{c}_q[j]$, is based on the Inverse Harmonic Mean (IHM) weight defined previously in [13] for speech coding,

$$\mathbf{c}_q[j] = \left[\frac{1}{\mathbf{x}_q[j] - \mathbf{x}_q[j-1]} + \frac{1}{\mathbf{x}_q[j+1] - \mathbf{x}_q[j]} \right] \quad (3)$$

where $\mathbf{x}_q[0] = 0$ and $\mathbf{x}_q[P+1] = \pi$ (assuming a Pth order stable LP analysis). Intuitively, since the weighting is inversely related to the distance between neighboring LSF parameters, mismatch in spectral peaks are weighed more heavily than mismatch in spectral valleys.

2.3. Clustering Method

The unit clustering algorithm is based on context–dependent monophone units with acoustic context classes previously formulated by Ljolje and Riley [14, 12]. Given sufficient training data, model parameters for triphone trajectories are estimated. However, a back–off model set of left–context, right–context, and context–independent units are also estimated. During data clustering, a training set of R observed trajectories is extracted from various examples of a particular phoneme (i.e., $\mathbf{X}^r$ for $r = 1, \ldots, R$). The clustering phase estimates the underlying parameters of a Q–state (synthesis) STM using the Linde–Buzo–Gray (LBG) algorithm with iterative centroid splitting,

1. Initialization : Initialize the number of modeled trajectories to 1 $(k = 1)$. For each state $(q = 1, \ldots, Q)$, compute the centroid mean vector μ_{qk} and diagonal covariance vector σ^2_{qk} from the sample mean and variance of the qth state of all available training tokens.
2. Splitting Phase : For each modeled trajectory $(k = 1, \ldots, K)$, split the trajectory if a sufficient number of training tokens exist. That is, the mean vector μ_{qk} is split into $\mu_{qk}(1 + \epsilon)$ and $\mu_{qk}(1 - \epsilon)$ where $\epsilon = 0.2\sigma_{qk}$.
3. Distortion Computation : Compute the distance of each training token to the current set of modeled trajectories. That is, for each training token, $\mathbf{X}^r$, compute $\mathrm{d}(\mathbf{X}^r, \mathbf{T}_k)$ given in Eq. 2 for $(k = 1, \ldots, K$; and $r = 1, \ldots, R)$.
4. Classification : With $\mathrm{d}(\mathbf{X}^r, \mathbf{T}_k)$, assign each training token to one of K current modeled trajectories such that the distortion function is minimized.
5. Model Update : Update the state–dependent mean vectors and variance terms of each modeled trajectory using the assigned observations from Step 4. Update the trajectory probability (p_k) as the count of training tokens assigned to the kth modeled trajectory divided by the total number of training tokens.
6. Iterate : Repeat Steps 2–5 until a convergence criterion is met or terminating iteration count is reached.

2.4. Model Parameter Based Synthesis

The synthesis algorithm is pitch–synchronous in nature. During processing, the left–context and right–context acoustic classes of the adjacent phonemes are determined and the most context sensitive model available is selected. Next, since the ith phoneme model consists of $K(i)$ possible parameter trajectories, a single trajectory must be selected to represent the current phoneme. Following [15], this is accomplished by conducting a Viterbi search for the trajectory sequence which minimizes a cost criteria. The best–path cost is comprised of a *selection cost* (related to how often the trajectory occurs in the training data) and a *concatenation cost* (related to the spectral discontinuity between two adjacent synthesis trajectories). Details of the search procedure and trajectory selection can be found in [8].

Once the appropriate trajectory sequence has been established, synthetic speech can be produced by first converting the LSF mean vector representing the qth model state into a corresponding LPC vector [11]. The vocal tract filter can then be excited by either a periodic pulse train during voiced speech or noise–like excitation during unvoiced speech. Since this simplified excitation model leads to poor

synthetic speech quality, we have considered tying an LP residual error waveform (extracted from the training data) to each trajectory model state in order to further convey speaker–dependent excitation. The tied LP residual to each modeled state is found by searching the short–time training data which minimizes the spectral distortion criterion given in Eq. 2. The PSOLA [10] analysis waveforms from the segment with minimal distance are decomposed into source and spectral envelope components. The corresponding Q–state LP residual sequence is then assigned to model the prototypical excitation for each state in the trajectory.

3. SPEAKER VERIFICATION EXPERIMENTS

3.1. YOHO Corpus Partitioning

The YOHO corpus [16] consists of 138 speakers (106 male, 32 female) producing short combination–lock phrases consisting of three doublets (e.g., "twenty–six, fifty–one, eighty–seven"). The doublets range in value from 21 to 99 with the following restrictions: (i) there are no exact decades (e.g., 20, 30, 40), (ii) there are no double digits (e.g., 44, 55), and (iii) there are no numbers ending in "8" (e.g., 28, 38). Because the vocabulary is restricted to doublet sequences, only 21 phonemes are present within the database.

Each speaker participated in 4 enrollment sessions consisting of 24 phrases each (i.e., $4 \times 24 = 96$ enrollment phrases). In addition, there are a total of 10 verification sessions, each of which consist of 4 phrases (i.e., $10 \times 4 = 40$ verification phrases). Since trajectory model estimation requires several minutes of training data, the experimental setup used in this section differs from the recommended database usage described in [16]. Specifically, it is necessary to partition the available data into three subsets to ensure open test evaluations: (i) data for estimating the LSF–STM synthesis units, (ii) data for estimating the GMM parameters for the speaker verifier, and (iii) data for imposter and customer trials. Therefore, the following data partition was considered,

1. 92 of the 96 enrollment phrases for each speaker are used to estimate the LSF–STM synthesis models.
2. 4 randomly selected enrollment phrases from each speaker were set aside for verification experiments.
3. 40 verification phrases are used to train the GMM–based speaker verification algorithm.

3.2. Baseline Speaker Verification Algorithm

Speaker verification can be described in terms of a two–hypothesis problem in which the verifier must decide whether the speech presented to the system was produced by the customer or by an imposter. Given an input sequence of T short–time speech feature vectors, $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, the hypothesis can be tested using the likelihood ratio,

$$\Lambda(\mathbf{O}) = \frac{p(\mathbf{O} \mid \mathcal{H}1)}{p(\mathbf{O} \mid \mathcal{H}0)} = \frac{p(\mathbf{O} \mid \lambda_c)}{p(\mathbf{O} \mid \lambda_{\bar{c}})}, \tag{4}$$

where λ_c and $\lambda_{\bar{c}}$ represent models for the customer and imposter respectively. Furthermore, the log–likelihood ratio can be expressed as,

$$\log \Lambda(\mathbf{O}) = \log p(\mathbf{O} \mid \lambda_c) - \log p(\mathbf{O} \mid \lambda_{\bar{c}}). \tag{5}$$

During processing the log–likelihood ratio is compared with a threshold, β, in order to decide hypothesis $\mathcal{H}0$ or $\mathcal{H}1$. For customer distributions modeled by GMMs, the observations are assumed statistically independent, therefore the log–likelihood of the observation sequence to the customer model is given by,

$$\log p(\mathbf{O} \mid \lambda_c) = \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{o}_t \mid \lambda_c). \tag{6}$$

The imposter model, $\lambda_{\bar{c}}$, in this study is comprised of a set of B background speaker models. The models include the $B/2$ nearest background speakers as well as $B/2$ farthest background speakers. Thus, each speaker enrolled in the system has a dedicated background model set. The normalizing term in Eq. 5 is then given by the log of the average likelihood across each of the background speaker models [1],

$$\log p(\mathbf{O} \mid \lambda_{\bar{c}}) = \log \left\{ \frac{1}{B} \sum_{b=1}^{B} p(\mathbf{O} \mid \lambda_b) \right\}. \tag{7}$$

The baseline speaker verifier was constructed in the following manner. First, 40 combination lock phrases found in the verification portion of the database were used to estimate a 32 mixture GMM for each customer (138 total). Observations consisting of 19 MFCCs were calculated every 10 msec and frames from silence regions were automatically discarded using an energy–based speech activity detector. Second, the background speaker model set was constructed by submitting the verification phrases from each customer to the 137 remaining GMMs. The 5 models with the largest log–probability and 5 models with the smallest log–probability were chosen as the near and far background set. Therefore, each customer score is normalized using Eq. 7 by ($B = 10$) background speakers.

Simulations were performed to find the distribution of the log–likelihood ratio output for each hypothesis ($\mathcal{H}1, \mathcal{H}0$). Considering the customer case, 4 enrollment combination–lock phrases were submitted to the corresponding customer model. Since there are 138 customers, there are ($138 \times 4 = 552$) values of log–$\Lambda(\mathbf{O})$ under hypothesis $\mathcal{H}1$. Likewise, 4 enrollment phrases from the remaining speakers in the database were used for imposter trials. Note that background speakers are excluded as imposters as suggested by Campbell [16]. Consequently, there are a total of ($138 \times 127 \times 4 = 70,104$) imposter tests for hypothesis $\mathcal{H}0$. Next, a decision threshold β, was varied to reveal a trade–off between false acceptance and false rejection errors. The baseline system achieved an EER of 1.45% ($\beta = 1.56$) for tests consisting of a single combination–lock phrase.

3.3. Voice–Altered Imposter Trials

While the false acceptance rate of the baseline system is low (1.45%), it is interesting to now consider the sensitivity of the speaker verifier to imposters whose speech has been transformed into the customer's voice characteristics prior to processing. To simulate this scenario, the proposed LSF–STM trajectory synthesis technique is used. During voice transformation, each combination–lock phrase is automatically segmented using an HMM–based segmenter [9]. The phoneme label sequence is obtained using a dictionary lookup for each digit doublet. The resulting phone sequence is submitted to the LSF–STM synthesis algorithm and a sequence of context–dependent LSF synthesis trajectories is determined. The input F_0 contour is left unmodified, however the median F_0 of the imposter is adjusted to match the median F_0 of the customer's voice using the PSOLA method.

Fig. 2 illustrates histogram plots of log–$\Lambda(\mathbf{O})$ under the different hypothesis scenarios. For Fig. 2A, the distribution of scores under hypothesis $\mathcal{H}1$ (customer access) is shown. Here, scores range in value from [+1,+6] while exhibiting an approximately Gaussian shaped distribution about the mean

value (approximately +3). The EER threshold ($\beta = 1.56$) is also shown as a solid line. Next, in Fig. 2B, the distribution of the log–likelihood scores under hypothesis $\mathcal{H}0$ (imposter access) are shown. Here, the majority of casual imposter attempts result in scores below the EER threshold suggesting a low false acceptance rate. The tail of the distribution is long resulting in scores ranging from -10 to values slightly greater than +2. The peak of the imposter distribution occurs at approximately +0.4 which is far below the EER threshold. Finally, in Fig. 2C, it is clear that voice alteration using LSF–STM synthesis impacts the verifier in 3 ways,

1. The overall peak in the imposter distribution is shifted from +0.4 to +2.2, a value *above* the EER threshold.
2. The range of imposter scores is reduced from approximately [-10,+2] to [0,+5]. The new distribution overlaps the customers' distribution under hypothesis $\mathcal{H}1$.
3. If the EER threshold is left unmodified, a substantial increase in false acceptance rate is noted. In fact, 86.1% of the altered–voice imposters using the LSF–STM synthesis scheme are falsely accepted by the verifier.

By increasing the decision threshold (β), a trade–off in customer false rejection versus imposter false acceptance can be determined. Table 1 summarizes the system performance for customer false rejection rates of 5%, 10%, 25%, and 50%. For example, at a false rejection rate of 25% (i.e., $\beta = 2.53$), 0.04% of the casual imposters and 34.6% of the voice–altered imposters are falsely accepted by the verifier. From this Table, it is clear that simply increasing the decision threshold yields an unacceptable customer false rejection rate.

FR	Threshold (β)	FA : (A)	FA : (B)
5%	1.87	0.43 %	72.2 %
10%	2.17	0.12 %	55.5 %
25%	2.53	0.04 %	34.6 %
50%	3.00	< 0.01 %	14.2 %

Table 1: Imposter false acceptance (FA %) for a given customer false rejection (FR %) rate. System performance is shown for (A) false acceptance for casual imposter attempts, and (B) false acceptance voice–altered imposter attempts.

4. DISCUSSION AND CONCLUSIONS

This paper has presented a new approach for trainable speech synthesis based on trajectory modeling of LSF parameters. A GMM–based verification algorithm was then constructed and shown to exhibit an EER 1.45% for casual imposter attempts. When the imposter voices are altered using the proposed synthesis algorithm, the false acceptance rate increases to 86% if the original EER decision threshold is left unmodified. However, it is worth considering the circumstances in which the results were obtained. First, synthesis models were estimated from phrases spoken by each customer. Therefore the domains of the training and testing material were matched. Furthermore, GMM–based verifiers lack the capability of confirming that the correct digit sequence was spoken by the imposter. However, we point out that the intelligibility of the synthetic speech digit sequences was found to be 99.5% in a formal listener evaluation. Furthermore, an additional listener evaluation of the processed synthetic speech confirms the similarity of the altered imposter voice to the customer voice (details can be found in [8]). Currently, we are considering conducting a more extensive evaluation using the NIST–SRE (1998) corpus.

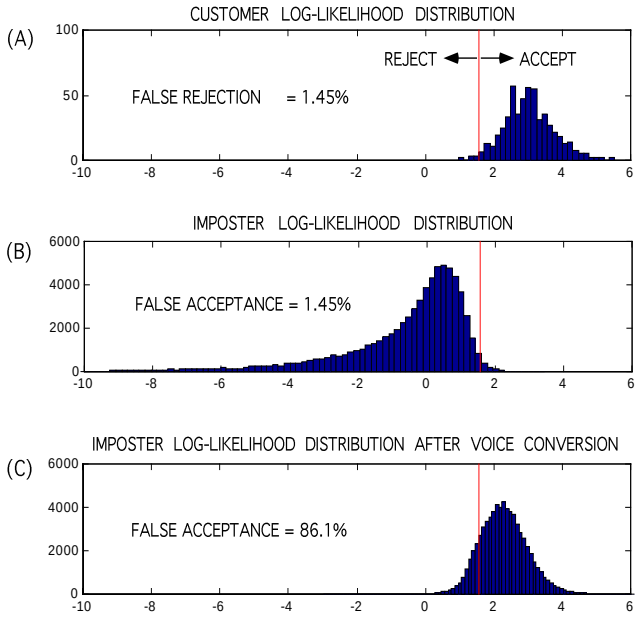

Figure 2: Histogram plot of log–likelihood ratio scores, $\Lambda(\mathbf{O})$, for (A) hypothesis $\mathcal{H}1$: customer access, (B) hypothesis $\mathcal{H}0$: casual imposter attempts, and for (C) hypothesis $\mathcal{H}0$: voice–altered imposter attempts.

5. REFERENCES

[1] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Comm.*, 17:91–108, 1995.

[2] A. Rosenberg and S. Parthasarathy. Speaker background models for connected digit password speaker verification. *ICASSP'96*, pp. 81–84.

[3] A. E. Rosenberg and M. R. Sambur. New techniques for automatic speaker verfication. *IEEE Trans. on Acoust., Speech, and Signal Process.*, ASSP-23(2):169–176, 1975.

[4] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *ICASSP'88*, pages 655–658, 1988.

[5] Y. Stylianou, O. Cappé, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. on Speech & Audio Process.*, 6(2):131–142, March 1998.

[6] Y. Gong and J.-P. Haton. Stochastic trajectory modeling for speech recognition. *ICASSP'94*, Vol. 1, pp. 57–60.

[7] Y. Gong. Stochastic trajectory modeling and sentence searching for continuous speech recognition. *IEEE Trans. on Speech & Audio Process.*, 5(1):33–44, January 1997.

[8] B. L. Pellom. *Enhancement, Segmentation, and Synthesis of Speech with Application to Robust Speaker Recognition.* PhD thesis, Duke University, September 1998.

[9] B. L. Pellom and J. H. L. Hansen. Automatic segmentation and labeling of speech recorded in unknown noisy channel environments. *Speech Comm.*, November 1998.

[10] E. Moulines and F. Charpentier. Pitch–synchronous waveform processing techniques for text–to–speech synthesis using diphones. *Speech Comm.*, 9:453–467, 1990.

[11] F. Soong and B.-H. Juang. Line spectrum pair and speech data compression. *ICASSP'84*, Vol. 1, pp. 1.10.1–1.10.4.

[12] A. Ljolje. High accuracy phone recognition using context clustering and quasi–triphonic models. *Comp. Speech and Lang.*, 8:129–151, 1994.

[13] R. Laroia, N. Phamdo, and N. Farvardin. Robust efficient quantization of speech LSP parameters using structured vector quantizers. *ICASSP'91*, pp. 641–644.

[14] A Ljolje and M.D. Riley. Automatic segmentation and labeling of speech. *ICASSP'91*, pp. 473–476.

[15] A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. *ICASSP'96*, Vol. 1, pp. 373–376.

[16] J. Campbell. Testing with the YOHO CD-ROM voice verification corpus. *ICASSP'95*, Vol. 1, pp. 341–344.