A MAXIMUM LIKELIHOOD APPROACH TO UNSUPERVISED ONLINE ADAPTATION OF STOCHASTIC VECTOR MAPPING FUNCTION FOR ROBUST SPEECH RECOGNITION

Donglai ZHU [1] *and Qiang HUO*[2]

[1] Institute for Infocomm Research, Singapore

[2]Department of Computer Science, The University of Hong Kong, Hong Kong, China

(E-mails: dzhu@i2r.a-star.edu.sg, qhuo@cs.hku.hk)

ABSTRACT

In the past several years, we've been studying feature transformation approaches for robust automatic speech recognition (ASR) based on the concept of stochastic vector mapping (SVM) to compensating for possible "distortions" caused by factors irrelevant to phonetic classification in both training and recognition stages. Although we have demonstrated the usefulness of the SVM-based approaches for several robust ASR applications where diversified yet representative training data are available, the performance improvement of SVM-based approaches is less significant when there is a severe mismatch between training and testing conditions. In this paper, we present a maximum likelihood approach to unsupervised online adaptation (OLA) of SVM function parameters on an utterance-by-utterance basis for achieving further performance improvement. Its effectiveness is confirmed by evaluation experiments on Finnish Aurora3 database.

Index Terms— robust speech recognition, online adaptation, feature compensation, maximum likelihood, hidden Markov model.

1. INTRODUCTION

Using feature transformation in training and/or recognition stages to compensate for possible "distortions" caused by factors irrelevant to phonetic classification has been studied in robust automatic speech recognition (ASR) area for many years. In the past several years, we've also been working on this research topic based on the concept of stochastic vector mapping (SVM) that performs a frame-dependent transformation to compensate for "environmental" variabilities in both training and recognition stages. We've studied several forms of SVM functions and two joint training approaches using maximum likelihood (ML) or minimum classification error (MCE) criteria respectively for the estimation of SVM and HMM parameters [14, 15, 17, 9].

There are several interesting works from other research groups that are related to our efforts. As discussed in [4], although the fMPE approach reported in [12] was derived with a different motivation, interestingly, its feature transformation is essentially the same as what was used in [14]. The main difference lies in the objective function used (MPE (minimum phone error) in [12] *vs* MCE in [14]) and the corresponding optimization procedures for training transformation and HMM parameters. The second work is the MMI-SPLICE approach (e.g., [5]) in which the objective function for parameter learning is maximum mutual information (MMI). The third work is the RDLT (Region Depedent Linear Transform) approach reported in [19], in which the piecewise linear transformations are applied to a vector concatenated from several frames of feature vectors and the training criterion is MPE.

Although we have demonstrated the usefulness of the SVM-based approaches for several robust ASR applications where diversified yet representative training data are available [14, 15, 17, 9], it was also observed that the performance improvement of SVM-based approaches is less significant when there is a severe mismatch between training and testing conditions. It is therefore natural to explore the idea of unsupervised online adaptation (OLA) of SVM parameters on an utterance-by-utterance basis and verify whether a further performance improvement can be achieved. The main purpose of this paper is to report our study on this topic. As a remark, the interesting works reported in [11, 10] are related to our work here, but both of them perform unsupervised online feature adaptation based on seed models (HMMs in [11] and GMMs/Eigenvoices in [10]) without feature compensation.

The rest of the paper is organized as follows. In Section 2, we summarize the SVM approaches to robust ASR. In Section 3, we present an ML formulation for OLA of SVM function parameters. Evaluation results on Finnish Aurora3 database are reported in Section 4. Finally, we conclude the paper in Section 5.

2. SVM APPROACHES

Let's assume that a speech utterance corrupted by some "distortions" has been transformed into a sequence of feature vectors. Given a set of training data $\mathcal{Y} = \{Y_i\}_{i=1}^{I}$, where Y_i is a sequence of feature vectors of original speech, suppose that they can be partitioned into E "environment" classes, and the D-dimensional feature vector y under an environment class e follows the distribution of a mixture of Gaussians, $p(y|e) = \sum_{k=1}^{K} p(k|e)p(y|k,e)$ $= \sum_{k=1}^{K} p(k|e)\mathcal{N}(y;\xi_k^{(e)}, R_k^{(e)})$, where $\mathcal{N}(\cdot;\xi,R)$ is a normal distribution with mean vector ξ and diagonal covariance matrix R. Readers are referred to [16] for the approach we used for the automatic clustering of environment conditions from training data $\mathcal{Y}$, the labeling of an utterance Y to a specific environment condition, and the estimation of the above model parameters. Given the set of Gaussian mixture models (GMM) $\{p(y|e)\}$, the task of frame-dependent SVM-based compensation is to estimate the compensated feature vector $\hat{x}$ from the original feature vector y by applying the environment-dependent transformation $\mathcal{F}(y;\Theta^{(e_y)})$, where $\Theta^{(e_y)}$ represents the trainable parameters of the transformation and e_y denotes the corresponding environment class to which y belongs. However, for the simplicity of notation, we will here-

This work was supported partially by grants from the RGC of the Hong Kong SAR, China.

inafter simply use e to denote the environment class to which y belongs, if no confusion will be caused according to the context.

So far we have studied five forms of SVM functions [14, 15, 17, 9]. The first one is borrowed from [3] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_1(y;\Theta^{(e)}) = y + \sum_{k=1}^{K} p(k|y,e) b_k^{(e)} \ , \tag{1}$$

where

$$p(k|y,e) = \frac{p(k|e)p(y|k,e)}{\sum_{j=1}^{K} p(j|e)p(y|j,e)} \ , \tag{2}$$

and $\Theta^{(e)} = \{b_k^{(e)}\}_{k=1}^{K}$. The second SVM function is borrowed from [2] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_2(y;\Theta^{(e)}) = y + b_k^{(e)} \ , \tag{3}$$

where, for the environment class e which y belongs to,

$$k = \arg \max_{k'=1,\ldots,K} p(k'|y,e) \ . \tag{4}$$

The third one is borrowed from [7] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_3(y;\Theta^{(e)}) = A^{(e)} y + b^{(e)} \ , \tag{5}$$

where $A^{(e)}$ is a nonsingular $D \times D$ matrix, $b^{(e)}$ is a D-dimensional vector, and $\Theta^{(e)} = \{A^{(e)}, b^{(e)}\}$. The fourth SVM function is defined in the form of piecewise linear transformations [9] and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_4(y;\Theta^{(e)}) = A^{(e)} y + \sum_{k=1}^{K} p(k|y,e) b_k^{(e)} \ , \tag{6}$$

where $\Theta^{(e)} = \{A^{(e)}; b_k^{(e)}, k = 1, \ldots, K\}$. The fifth SVM function [9] is similar to Eq. (6) and listed as follows:

$$\hat{x} \triangleq \mathcal{F}_5(y;\Theta^{(e)}) = A^{(e)} y + b_k^{(e)} \ , \tag{7}$$

where k is calculated by using Eq. (4).

Let's assume that each basic speech unit in our speech recognizer is modeled by a Gaussian mixture continuous density HMM (CDHMM), whose parameters are denoted as $\lambda = \{\pi_s, a_{ss'}, c_{sm}, \mu_{sm}, \Sigma_{sm}; s, s' = 1, \cdots, S; m = 1, \cdots, M\}$, where S is the number of states, M is the number of Gaussian components for each state, $\{\pi_s\}$ is the initial state distribution, $a_{ss'}$'s are state transition probabilities, c_{sm}'s are Gaussian mixture weights, $\mu_{sm} = [\mu_{sm1}, \cdots, \mu_{smD}]^{Tr}$ is a D-dimensional mean vector, and $\Sigma_{sm} = diag\{\sigma_{sm1}^2, \cdots, \sigma_{smD}^2\}$ is a diagonal covariance matrix. Our environment compensated training approach is to adjust SVM function parameters $\Theta = \{\Theta^{(e)}, e = 1, \cdots, E\}$ and CDHMM parameters $\Lambda = \{\lambda\}$ to optimize a training objective function. For example, the ML training approaches of $\mathcal{F}_1$ and $\mathcal{F}_2$ are presented in [15, 17]. The ML training approaches of $\mathcal{F}_3$ and $\mathcal{F}_4$ are presented in [9]. The ML training procedure of $\mathcal{F}_5$ is similar to that of $\mathcal{F}_4$, where the only difference is that the training feature vectors are compensated with Eq. (7) rather than Eq. (6) before the estimation of CDHMM parameters Λ.

In recognition, given an unknown utterance Y, the most similar training environment class e is identified first (e.g. [16]). Then, the corresponding GMM and the mapping function are used to derive a compensated version $\hat{X}$ from Y. For the convenience of notation, we also use hereinafter $\mathcal{F}(Y;\Theta^{(e)})$ to denote the compensated version of the utterance Y by transforming individual feature vector y_t as defined in the previous SVM functions. After feature compensation, $\hat{X}$ is finally recognized by an HMM-based recognizer trained as described in [14] or [17] or [9].

3. ONLINE ADAPTATION OF SVM PARAMETERS

For "unseen" distortions that are not covered in training conditions but exist in testing conditions, the pre-trained SVM parameters may not work as effectively as expected. To mitigate the problem, one solution is to perform an unsupervised online adaptation (OLA) using the utterance to be recognized to adapt the SVM parameters to characterize the new environment better. Apparently, there are many ways of doing OLA. As a first step, we tried a simple ML approach that maximizes the following likelihood function defined on the testing utterance Y by adjusting SVM parameters Θ:

$$\mathcal{L}(\Theta) = p(\mathcal{F}(Y;\Theta)|\Lambda). \tag{8}$$

Among the five SVM functions, $\mathcal{F}_1$ and $\mathcal{F}_2$ belong to bias removal techniques, while $\mathcal{F}_4$ and $\mathcal{F}_5$ belong to piecewise linear transformations techniques. $\mathcal{F}_3$ is not frame-dependent compensation and thus is not studied in this paper. We choose two SVM functions $\mathcal{F}_2$ and $\mathcal{F}_5$ for OLA study due to the following reasons:

- $\mathcal{F}_1$ and $\mathcal{F}_2$ achieve similar performance according to experimental results reported in [17]. So do $\mathcal{F}_4$ and $\mathcal{F}_5$ according to results shown in Section 4;
- In comparison with $\mathcal{F}_1$ and $\mathcal{F}_4$, $\mathcal{F}_2$ and $\mathcal{F}_5$ have relatively simple derivation and low computational complexity;
- $\mathcal{F}_2$ and $\mathcal{F}_5$ use consistent transformation functions in both training and recognition stages, while $\mathcal{F}_1$ and $\mathcal{F}_4$ use hybrid approaches in two stages [17, 9].

3.1. Online Adaptation of $\mathcal{F}_2$

Given a set of $\mathcal{F}_2$ parameters $\Theta = \{b_k^{(e)}; k = 1, \ldots, K; e = 1, \ldots, E\}$ and CDHMM parameters Λ that are estimated from the training data using ML SVM approach, the OLA problem here is to adjust Θ to maximize the likelihood function in Eq. (8). The updating formula of $b_k^{(e)}$ can be derived easily by using EM algorithm as follows:

$$b_{kd}^{(e)} = \frac{\sum_{t,s,m} \mathbf{1}[k,t]\zeta_t(s,m)(\mu_{smd} - y_{td})/\sigma_{smd}^2}{\sum_{t,s,m} \mathbf{1}[k,t]\zeta_t(s,m)/\sigma_{smd}^2} \ , \tag{9}$$

where

$$\mathbf{1}[k,t] = \begin{cases} 1 \text{ if } k = \arg\max_{k'} p(k'|y_t,e) \\ 0 \text{ otherwise} \end{cases} . \tag{10}$$

In the above equation, $\zeta_t(s,m)$ is the occupation probability of Gaussian component m in state s, at time t of the current compensated observation $\hat{x}_t = \mathcal{F}_2(y_t;\Theta^{(e)})$. It can be calculated with a Forward-Backward procedure using the compensated utterance $\hat{X}$ against the CDHMM Λ.

Therefore, the unsupervised online adaptation procedure of $\mathcal{F}_2$ includes the following steps:

Step 1. Given an unknown utterance Y, the most similar training environment class e is identified first (e.g. [16]). Then Y is converted to a compensated version $\hat{X}$ by using $\mathcal{F}_2$. The compensated utterance $\hat{X}$ is then recognized via Viterbi decoding with pre-trained CDHMM parameters Λ.

Step 2. Given the recognized transcription, $b_k^{(e)}$'s are updated by using Eq. (9).

Step 3. The utterance Y is converted to a compensated version $\hat{X}$ by using $\mathcal{F}_2$ with the updated parameters $\{b_k^{(e)}\}$. Then $\hat{X}$ is recognized with the pre-trained CDHMM parameters Λ again.

Step 4. Steps 2 and 3 can be repeated until a pre-specified criterion is satisfied (e.g., a fixed number of cycles).

3.2. Online Adaptation of $\mathcal{F}_5$

The parameter set of $\mathcal{F}_5$ is $\Theta = \{A^{(e)}, b_k^{(e)}; k = 1, \ldots, K; e = 1, \ldots, E\}$. We may adapt both $A^{(e)}$ and $b_k^{(e)}$ to the current utterance to be recognized. However, our preliminary results show that adaptation of $A^{(e)}$ encounters numerical problem because of sparse adaptation data. A possible way to address this issue is to use Bayesian estimation rather than the ML estimation, which will be a topic of our future work. Therefore, we only discuss the adaptation of $b_k^{(e)}$ in this paper. Again, by using the EM algorithm, the updating formula of $b_k^{(e)}$ in $\mathcal{F}_5$ can be derived as follows:

$$b_{kd}^{(e)} = \frac{\sum_{t,s,m} \mathbf{1}[k,t]\zeta_t(s,m)(\mu_{smd} - A_d^{(e)} \cdot y_t)/\sigma_{smd}^2}{\sum_{t,s,m} \mathbf{1}[k,t]\zeta_t(s,m)/\sigma_{smd}^2}, \quad (11)$$

where $\mathbf{1}[k,t]$ is calculated with Eq. (10), and $A_d^{(e)}$ is the dth row of $A^{(e)}$. Note that $\zeta_t(s,m)$ is the occupation probability of Gaussian component m in state s, at time t of the current compensated observation $\hat{x}_t = \mathcal{F}_5(y_t; \Theta^{(e)})$.

The OLA procedure for $\mathcal{F}_5$ is similar to that of $\mathcal{F}_2$. The difference is to use $\mathcal{F}_5$ rather than $\mathcal{F}_2$ in the relevant steps.

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

We use Finnish Aurora3 database [1] to verify our algorithm. Aurora3 contains utterances of connected digits that were recorded by using both close-talking (CT) and hands-free (HF) microphones in cars under several driving conditions to reflect some realistic scenarios for typical in-vehicle ASR applications. There are roughly three conditions: *quiet*, *low noise*, and *high noise*. The database is divided into following three subsets according to matching degree between training data and test data:

- **Well-Matched (WM) condition**: Both training and testing data include utterances recorded by both CT and HF microphones from all conditions;
- **Medium-Mismatched (MM) condition**: Training data includes utterances recorded by HF microphone in the *quiet* and *low noise* conditions. Testing data includes utterances recorded by HF microphone in the *high noise* condition;
- **High-Mismatched (HM) condition**: Training data includes utterances recorded by CT microphone from all conditions. Testing data includes utterances recorded by HF microphone in the *low noise* and *high noise* conditions.

Therefore, the MM condition simulates mainly the mismatch caused by a noisy environment due to different driving speeds and possible background music. The HM condition simulates mainly the mismatch caused by different transducers.

In our experiments, the ETSI Advanced Front-End (AFE) as described in [6] is used for feature extraction from a speech utterance. A feature vector sequence is extracted from the input speech utterance via a sequence of processing modules that include noise reduction, waveform processing, cepstrum calculation, blind equalization, and "server feature processing". Each frame of feature vector has 39 features that consists of 12 MFCCs (C_1 to C_{12}), a combined log energy and C_0 term, and their first and second order derivatives. Although all the feature vectors are computed from a given speech utterance, the feature vectors that are sent to the speech recognizer and the training module are those corresponding to speech frames, as detected by a VAD module described in Annex A of [6]. In SVM-based experiments, all the training data are clustered into 8 different environment classes (i.e. $E = 8$), of which each is modeled by a GMM consisting of 32 Gaussian components (i.e. $K = 32$).

Each digit is modeled as a whole word left-to-right HMM with 16 emitting states, 3 Gaussian mixture components with diagonal covariance matrices per state. Besides, two pause models, "sil" and "sp", are created to model the silence before/after the digit string and the short pause between any two digits, respectively. The "sil" model is a 3-emitting state HMM with a flexible transition structure as described in [8]. Each state is modeled by a mixture of 6 Gaussian components with diagonal covariance matrices. The "sp" model consists of 2 dummy states and a single emitting state which is tied with the middle state of "sil". During recognition, an utterance can be modeled by any sequence of digits with the possibility of a "sil" model at the beginning and at the end and a "sp" model between any two digits. Recognition experiments are performed with the search engine of HTK toolkit [18].

For the convenience of reference, we have used the terms of SVM1, SVM2, SVM3, and SVM4 to refer to different SVM-based approaches in [17, 9]. Here, we define one more approach, SVM5, in which the SVM function $\mathcal{F}_5(y; \Theta^{(e)})$ in Eq. (7) is used in both training and recognition for feature compensation.

Table 1. A comparison of word error rates (in %) of five SVM-based approaches versus the CDHMM baseline system without feature compensation.

Methods	WM ($\times 40\%$)	MM ($\times 35\%$)	HM ($\times 25\%$)	Average
Baseline	3.95	19.70	14.28	12.05
SVM1	3.33	17.78	16.01	11.56
SVM2	3.34	17.58	16.15	11.53
SVM3	3.08	16.48	15.37	10.84
SVM4	2.92	16.62	16.71	11.16
SVM5	2.92	16.48	16.61	11.09

4.2. Results of Different Baseline Systems

Table 1 summarizes a comparison of word error rates (WERs in %) of five SVM-based approaches versus the CDHMM baseline system without feature compensation. It is observed that the SVM approaches achieve better performance than the CDHMM baseline system for the WM and MM conditions, but degrade the performance for the HM condition. It indicates that the pre-trained SVM functions cannot effectively compensate for distortions in highly mismatched testing data. SVM1 and SVM2 achieve similar performance, so do SVM4 and SVM5. Therefore, rather than doing online adaptation based on all SVM functions, we adopt SVM2 and SVM5 for further study of online adaptation.

Table 2. A comparison of word error rates (in %) of the CDHMM-based baseline system, the SVM2-based baseline system, and the adapted SVM2-based systems with different OLA cycles.

Testing Conditions	CDHMM Baseline	SVM2 Baseline	OLA Cycles	
			1	2
WM($\times 40\%$)	3.95	3.34	3.26	3.26
MM($\times 35\%$)	19.70	17.58	16.14	15.87
HM($\times 25\%$)	14.28	16.15	13.64	11.98
Average	12.05	11.53	10.36	9.85

Table 3. A comparison of word error rates (in %) of the CDHMM-based baseline system, the SVM5-based baseline system, and the adapted SVM5-based systems with different OLA cycles.

Testing Conditions	CDHMM Baseline	SVM5 Baseline	OLA Cycles	
			1	2
WM($\times 40\%$)	3.95	2.92	2.73	2.73
MM($\times 35\%$)	19.70	16.48	14.84	14.43
HM($\times 25\%$)	14.28	16.61	13.71	12.54
Average	12.05	11.09	9.71	9.28

4.3. Results of Unsupervised Online Adaptation

Unsupervised online adaptation (OLA) of SVM2 or SVM5 functions is performed on each testing utterance according to the procedures described in Section 3 for two adaptation cycles. Tables 2 and 3 summarize WERs of the adapted systems based on SVM2 and SVM5, respectively. For comparison, we also list the results of the CDHMM-based baseline system without feature compensation and the corresponding SVM-based baseline system. It is observed that unsupervised OLA can indeed improve the performance further. We have also conducted a comparative study with two existing robust ASR approaches in literature: unsupervised MLLR (e.g. [7]) and feature-space stochastic matching (SM) [13]. For each testing utterance, a global diagonal transformation matrix and a bias vector are estimated in MLLR, while a bias vector is estimated in SM approach. Two adaptation cycles are performed for both approaches. The performance of different OLA approaches is compared in Table 4. The SVM5-based approach achieves the best overall performance.

5. SUMMARY

In this paper, we have studied an ML approach to unsupervised online adaptation (OLA) of two SVM functions: SVM2 and SVM5. Evaluation results on Finnish Aurora3 database show that in comparison with the CDHMM-based baseline system, unsupervised OLA of SVM2 yields a relative word error rate reduction of 17.5%, 19.4% and 16.1% for WM, MM and HM conditions respectively, while unsupervised OLA of SVM5 yields a relative word error rate reduction of 30.9%, 26.8%, 12.2% respectively.

Table 4. A comparison of word error rates (in %) of several unsupervised online adaptation (OLA) approaches versus the CDHMM baseline system without feature compensation.

Methods	WM ($\times 40\%$)	MM ($\times 35\%$)	HM ($\times 25\%$)	Average
Baseline	3.95	19.70	14.28	12.05
MLLR	4.01	23.05	8.76	11.86
SM	3.65	16.48	11.27	10.05
SVM2-OLA	3.26	15.87	11.98	9.85
SVM5-OLA	2.73	14.43	12.54	9.28

6. REFERENCES

[1] Aurora document AU/217/99, "Availability of Finnish speechdat-car database for ETSI STQ WI008 front-end standardisation," Nokia Nov 1999.

[2] L. Deng, A. Acero, M. Plumpe, and X.-D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," *Proc. ICSLP 2000*, pp.III–806–809.

[3] L. Deng, A. Acero, L. Jiang, J. Droppo, and X.-D. Huang, "High-performance robust speech recognition using stereo training data," *Proc. ICASSP 2001*, pp. 301–304.

[4] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 477–480, 2005.

[5] J. Droppo and A. Acero, "Joint discriminative front end and back end training for improved speech recognition accuracy," *Proc. ICASSP 2006*, pp. 281–284.

[6] ETSI standard document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," ETSI ES 202 050 v1.1.1 (2002-10), 2002.

[7] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[8] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *ISCA ITRW ASR-2000*, Paris, France, 2000.

[9] Q. Huo and D. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," *Proc. Interspeech 2006 – ICSLP*, pp.1129-1132.

[10] P. Kenny, V. Gupta, G. Boulianne, P. Ouellet, and P. Dumouchel, "Feature normalization using smoothed mixture transformations," *Proc. Interspeech 2006 – ICSLP*, pp.25-28.

[11] S. S. Kozat, K. Visweswariah, and R. Gopinath, "Feature adaptation based on Gaussian posteriors," *Proc. ICASSP 2006*, pp. 221–224.

[12] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," *Proc. ICASSP 2005*, pp. 961–964.

[13] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 3, pp.190-202, 1996.

[14] J. Wu and Q. Huo, "An environment compensated minimum classification error training approach and its evaluation on AURORA2 database," *Proc. ICSLP 2002*, pp. 453–456.

[15] J. Wu and Q. Huo, "Several HKU approaches for robust speech recognition and their evaluation on AURORA connected digit recognition tasks," *Proc. Eurospeech 2003*, pp. 21–24.

[16] J. Wu, D. Zhu, and Q. Huo, "A study of minimum classification error training for segmental switching linear Gaussian hidden Markov models," *Proc. ICSLP 2004*.

[17] J. Wu, Q. Huo, and D. Zhu, "An environment compensated maximum likelihood training approach based on stochastic vector mapping," *Proc. ICASSP 2005*, pp. 429–432.

[18] S. J. Young, *et al.*, *The HTK Book* (for HTK Version 3.3), 2005.

[19] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," *Proc. ICASSP 2006*, pp. 313–316.