Factored MLLR Adaptation For Singing Voice Generation

June Sig Sung, Doo Hwa Hong, Shin Jae Kang and Nam Soo Kim

School of Electrical Engineering and INMC
Seoul National University, Korea
{jssung, dhhong, sjkang}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

In our previous study, we proposed factored MLLR (FMLLR) where each MLLR parameter is defined as a function of a control vector. We presented a method to train the FMLLR parameters based on a general framework of the expectation-maximization (EM) algorithm. In this paper, we extend the FMLLR structure from diagonal to unrestricted full matrix with a sophisticated algorithm for the training of relevant parameters. In the experiments on artificial generation of singing voice, we evaluate the performance of the FMLLR technique with two matrix structures and also compare with other approaches to parameter adaptation in HMM-based speech synthesis.

Index Terms: Parameter adaptation, MLLR, HRHSMM, factored MLLR

1. Introduction

Maximum likelihood linear regression (MLLR) is one of the most popular techniques for parameter adaptation in hidden Markov model (HMM)-based systems [1]. In the MLLR approach, original parameters of the HMM-based system are mapped to their adapted values via a set of affine transformations which are estimated from a small amount of adaptation data. MLLR was first proposed for speaker adaptation in order to improve the performance of the speech recognition systems, and later a variety of extensions have been developed with applications to other areas [2]-[6].

Generally in MLLR adaptation, the regression parameters are shared among a group of speech units in order to achieve robust parameter estimation with limited amount of data. This parameter sharing is usually implemented via a decision tree where each node is associated with a binary question that distributes the incoming speech units into two child nodes. The binary questions are selected from a finite number of questions concerned with lexical, contextual and other suprasegmental conditions. However, it becomes practically impossible for the MLLR technique to be applied to parameter adaptation when we need separate regression parameters for a huge number of conditions. For instance in singing voice synthesis, it is desired to apply different adaptation parameters depending on the given musical note. In the case of expressive speech synthesis, it is sometimes requested to control the expressive level or intensity in a continuous scale.

In order to alleviate the difficulty of the conventional MLLR approach, Nose et al. proposed the multiple regression hidden semi-Markov model (MRHSMM) [7]-[9]. In the MRHSMM technique, each HMM mean vector is described by performing multiple regression on a low dimensional style vector where each component represents the degree of intensity of a specific style. By varying the style vector, the expressivity of each emotional state can be controlled in the synthesized speech. With MRHSMM, an HMM mean vector is given by a linear combination of a number of basis vectors where each basis vector corresponds to a specific speaking style. It has been reported that MRHSMM provides an efficient way to model and control the intensity of several emotional expressions and speaking styles that appear in natural speech.

Even though MRHSMM is efficient for a flexible representation of HMM parameters, its application would be problematic when the number of typical speaking styles becomes huge or even infinite. Particularly for the singing voices, it is known that the vocal tract configuration varies depending not only on the phonetic information but also on the musical notes which provide the information concerned with tone and rhythm [10]. If we apply the conventional MLLR technique to adapt the parameters of a reading-style speech synthesizer to singing voices, we should compute separate transforms for different musical notes. Moreover, since the number of styles to consider is huge, the MRHSMM-type algorithm is not suitable for this application.

In our previous work, we extended the conventional MLLR to the factored MLLR (FMLLR) framework where each MLLR parameter is defined as a function of the control parameter vector [11]. More specifically, each element of the MLLR parameters is given as an inner product between a regression vector and a transformed control vector. This approach is motivated by the well-known fact that a complicated non-linear function can be efficiently approximated by an inner product in a higher dimensional feature space [12]. To show the effectiveness, we applied the FMLLR approach to adapt the spectral envelope features of the reading-style speech to those of the singing voice and compared its performance with the traditional MLLR approaches.

In this paper, we further improve the performance of FMLLR by extending its structure from diagonal to full matrix. We also present a training method to estimate the full matrix FMLLR parameters based on the expectation-maximization (EM) algorithm. Performance of the proposed technique is evaluated on experiments on singing voice synthesis, and compared with the conventional MLLR, MRHSMM and diagonal structured FMLLR methods.

2. Factored MLLR

MLLR computes a set of affine transforms aiming at reducing the mismatch between the base model and the given adaptation data. The affine transforms are applied to adapt the mean and covariance of each Gaussian in the HMM system. These transforms make it possible to relocate the mean vector components and alter the covariances of the base model so that each state of the HMM system is more likely to generate the adaptation data. In conventional MLLR adaptation, the mean vector $\mu_s \in R^p$ of

28 – 31 August 2011, Florence, Italy

a particular distribution s of the HMM is transformed to $\hat{\mu}_s$ via

$$\hat{\mu}_s \quad = \quad \mathbf{M}\mu_s + \mathbf{b} \tag{1}$$

where $\mathbf{M}$ is a $p \times p$ regression matrix and $\mathbf{b}$ is a bias vector. The output probability density function (PDF) of the distribution s is assumed to be a single Gaussian with the mean vector μ_s and covariance matrix Σ_s.

Now suppose that for a particular purpose $\mathbf{M}$ and $\mathbf{b}$ should depend on a control parameter η which is generally a continuous-valued vector of dimension D. This implies that the mean vector of the distribution s is adapted differently depending on η. Under this framework, (1) is rewritten as

$$\hat{\mu}_s \quad = \quad \mathbf{M}\left(\eta\right)\mu_s + \mathbf{b}\left(\eta\right) \tag{2}$$

and this approach is called the FMLLR technique [11].

One of the most important issues in the FMLLR approach is how to estimate $\mathbf{M}\left(\eta\right)$ and $\mathbf{b}\left(\eta\right)$ from a set of adaptation data. Even with a large amount of adaptation data it is impractical to estimate $\mathbf{M}\left(\eta\right)$ and $\mathbf{b}\left(\eta\right)$ for each possible value of η separately. For that reason, we need a parametric form for $\mathbf{M}\left(\eta\right)$ and $\mathbf{b}\left(\eta\right)$ of which parameters can be estimated efficiently based on the given adaptation data.

In this section, we propose a technique that enables a parametric representation of $\mathbf{M}\left(\eta\right)$ and $\mathbf{b}\left(\eta\right)$, and present an algorithm to estimate the relevant parameters with diagonal and full matrix structures for the regression matrix $\mathbf{M}$. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$ be the given adaptation data vectors. Different from conventional MLLR adaptation, now each adaptation vector $\mathbf{x}_t$ is accompanied with the corresponding control parameter η_t.

2.1. Diagonal FMLLR

In this case, the regression matrix $\mathbf{M}\left(\eta\right)$ and the covariance matrix Σ_s are assumed to be diagonal, and represented in the following parametric form:

$$\mathbf{M}\left(\eta\right) \quad = \quad \text{diag}\left(\mathbf{w}_1'\xi,\ \mathbf{w}_2'\xi,\ \cdots,\ \mathbf{w}_p'\xi\right) \tag{3}$$

$$\mathbf{b}\left(\eta\right) \quad = \quad \left(\mathbf{v}_1'\xi,\ \mathbf{v}_2'\xi,\ \cdots,\ \mathbf{v}_p'\xi\right)' \tag{4}$$

where $\xi = \phi\left(\eta\right)$ is an L-dimensional control vector obtained by transforming the control parameter η. In (3) and (4), both $\mathbf{W}_d = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_p\}$ and $\mathbf{V}_d = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\}$ are L-dimensional regression vectors which are the core parameters of FMLLR. Note that subscript d denotes that $\mathbf{M}$ is a diagonal matrix. This approach is motivated by the techniques popular in machine learning community, where an underlying classification or regression function is approximated by an inner product in a higher dimensional feature space [12]. Even though (3) and (4) can be possibly extended to the well-known kernel methods, in this work, we will use the original inner product form.

In order to estimate $\mathbf{W}_d$ and $\mathbf{V}_d$, we follow the EM algorithm employed in the conventional MLLR technique. At the *E* (expectation) step of the EM algorithm, we compute the a posteriori probability of the distribution s at each time defined by

$$\gamma_t\left(s\right) \quad = \quad Pr\left(\theta(t) = s|\mathbf{X},\ \lambda\right) \tag{5}$$

where $\theta(t)$ indicates the distribution index at time t and λ represents the current adaptation parameters. After the posterior probability $\gamma_t\left(s\right)$ is computed, we update the parameters $\mathbf{W}_d$ and $\mathbf{V}_d$ according to

$$\left\{\widehat{\mathbf{W}}_d,\ \widehat{\mathbf{V}}_d\right\} = \arg\max_{\{\mathbf{W}_d,\ \mathbf{V}_d\}} \mathcal{L}\left(\mathbf{W}_d,\ \mathbf{V}_d\right) \tag{6}$$

where

$$\mathcal{L}\left(\mathbf{W}_d,\ \mathbf{V}_d\right) = -\frac{1}{2}\sum_{t=1}^{T}\gamma_t\left(s\right) \times\left(\sum_{i=1}^{p}\frac{\left(x_{t,i} - \mathbf{w}_i'\xi_t\mu_{s,i} - \mathbf{v}_i'\xi_t\right)^2}{\sigma_{s,i}^2}\right),$$

in which $\widehat{\mathbf{W}}_d$ and $\widehat{\mathbf{V}}_d$ are the updated parameters for diagonal FMLLR and $\xi_t = \phi\left(\eta_t\right)$ is the transformed control vector accompanied to $\mathbf{x}_t$. The solution to (6) is obtained by setting the gradients of $\mathcal{L}\left(\mathbf{W}_d,\ \mathbf{V}_d\right)$ with respect to $\mathbf{W}_d$ and $\mathbf{V}_d$ to zero. After some manipulations for each dimension i, we are led to

$$\begin{bmatrix} \left(\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,i}^2}{\sigma_{s,i}^2}\xi_t\xi_t'\right) & \left(\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,i}}{\sigma_{s,i}^2}\xi_t\xi_t'\right) \\ \left(\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,i}}{\sigma_{s,i}^2}\xi_t\xi_t'\right) & \left(\sum_{t=1}^{T}\gamma_t(s)\frac{1}{\sigma_{s,i}^2}\xi_t\xi_t'\right) \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{w}}_i \\ \widehat{\mathbf{v}}_i \end{bmatrix}$$

$$= \begin{bmatrix} \left(\sum_{t=1}^{T}\gamma_t(s)\frac{x_{t,i}\mu_{s,i}}{\sigma_{s,i}^2}\xi_t\right) \\ \left(\sum_{t=1}^{T}\gamma_t(s)\frac{x_{t,i}}{\sigma_{s,i}^2}\xi_t\right) \end{bmatrix} \tag{7}$$

where $\widehat{\mathbf{w}}_i$ and $\widehat{\mathbf{v}}_i$ are the updated parameters for $\mathbf{w}_i$ and $\mathbf{v}_i$, respectively.

2.2. Full matrix FMLLR

Now suppose that $\mathbf{M}(\eta)$ is a full matrix as given by

$$\mathbf{M}(\eta) = \begin{pmatrix} \mathbf{w}_{11}'\xi & \mathbf{w}_{12}'\xi & \cdots & \mathbf{w}_{1p}'\xi \\ \mathbf{w}_{21}'\xi & \mathbf{w}_{22}'\xi & \cdots & \mathbf{w}_{2p}'\xi \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{p1}'\xi & \mathbf{w}_{p2}'\xi & \cdots & \mathbf{w}_{pp}'\xi \end{pmatrix} \tag{8}$$

where $\mathbf{w}_{ij}$ is an L-dimensional vector needed for the (i,j)-th component of $\mathbf{M}(\eta)$. Similar to diagonal FMLLR, we follow the strategy of the EM algorithm. Considering (8), we can rewrite (2) component-wisely

$$\hat{\mu}_{s,i} = \sum_{j=1}^{p}\mathbf{w}_{ij}'\xi\mu_{s,j} + \mathbf{v}_i'\xi. \tag{9}$$

Based on this, parameter estimation is performed according to

$$\left\{\widehat{\mathbf{W}}_f,\ \widehat{\mathbf{V}}_f\right\} = \arg\max_{\{\mathbf{W}_f,\ \mathbf{V}_f\}} \mathcal{L}\left(\mathbf{W}_f,\ \mathbf{V}_f\right) \tag{10}$$

where

$$\mathcal{L}(\mathbf{W}_f, \mathbf{V}_f) = -\frac{1}{2}\sum_{t=1}^{T}\gamma_t(s) \times\left(\sum_{i=1}^{p}\frac{\left(x_{t,i} - \sum_{j=1}^{p}\mathbf{w}_{ij}'\xi_t\mu_{s,j} - \mathbf{v}_i'\xi_t\right)^2}{\sigma_{s,i}^2}\right)$$

with the subscript f denoting the full matrix FMLLR case. The solution to (10) is obtained by setting the gradients of $\mathcal{L}\left(\mathbf{W}_f,\ \mathbf{V}_f\right)$ to zero. After some manipulations for each dimension i, we have an equation shown in (11).

$$
\begin{pmatrix}
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}^2}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}\mu_{s,2}}{\sigma_{s,i}^2}\xi_t\xi_t' & \cdots & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}\mu_{s,p}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}}{\sigma_{s,i}^2}\xi_t\xi_t' \\
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}\mu_{s,1}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}^2}{\sigma_{s,i}^2}\xi_t\xi_t' & \cdots & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}\mu_{s,p}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}}{\sigma_{s,i}^2}\xi_t\xi_t' \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,p}\mu_{s,1}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,p}\mu_{s,2}}{\sigma_{s,i}^2}\xi_t\xi_t' & \cdots & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,p}^2}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,p}}{\sigma_{s,i}^2}\xi_t\xi_t' \\
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}}{\sigma_{s,i}^2}\xi_t\xi_t' & \cdots & \sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,p}}{\sigma_{s,i}^2}\xi_t\xi_t' & \sum_{t=1}^{T}\gamma_t(s)\frac{1}{\sigma_{s,i}^2}\xi_t\xi_t'
\end{pmatrix}
\begin{pmatrix}
\widehat{\mathbf{w}}_{i1} \\ \widehat{\mathbf{w}}_{i2} \\ \vdots \\ \widehat{\mathbf{w}}_{ip} \\ \widehat{\mathbf{v}}_i
\end{pmatrix}
$$

$$
= \begin{pmatrix}
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,1}x_{t,i}}{\sigma_{s,i}^2}\xi_t \\
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,2}x_{t,i}}{\sigma_{s,i}^2}\xi_t \\
\vdots \\
\sum_{t=1}^{T}\gamma_t(s)\frac{\mu_{s,o}x_{t,i}}{\sigma_{s,i}^2}\xi_t \\
\sum_{t=1}^{T}\gamma_t(s)\frac{x_{t,i}}{\sigma_{s,i}^2}\xi_t
\end{pmatrix} \tag{11}
$$

3. Experiments

The objective in this experiment is to apply the transform methods presented in Section 2 to adapt the parameters of a reading-style speech synthesizer to a set of given singing voice. Since it is difficult to collect a large amount of singing voice and reading-style speech simultaneously from the same speaker, we attempted to transform the parameters of a reading-style speech synthesizer with a small amount of singing voice data. For the construction of reading-style speech synthesizers, we used the reading-style speech data spoken by two female speakers: YMK and SJK. The speaker YMK provided only the reading-style speech data while both the reading-style and singing voice data were available for the speaker SJK. The reading-style speech synthesizer for the speaker YMK was trained with 4,000 utterances amounting to 525 minutes. On the other hand, the reading-style speech synthesizer for the speaker SJK was obtained by adapting the parameters of the speaker YMK with 162 utterances amounting to 32 minutes.

Each utterance was sampled at 16 kHz and a 20 ms Hamming window was applied with 5 ms frame shift for speech feature extraction. As for the spectrum feature, a 25th-order mel-scaled cepstrum vector was extracted at each frame. By attaching the Δ- and $\Delta\Delta$-cepstra derived from the extracted mel-scaled cepstrum sequence, the spectrum feature could be represented by a 75-dimensional vector at each frame. We also extracted the pitch from each frame for the generation of voiced excitation signals. As the basic unit of speech synthesis, we applied triphones and each triphone was modeled by a 5 state left-to-right structured HMM where the observation distribution at each state was given by a single Gaussian PDF with diagonal covariance matrix.

The parameters of the HMM-based speech synthesizer for the reading-style speech were trained by following the general technique presented in [13]. For a robust parameter estimation, the decision tree technique was employed to share the observation distributions across the states, which resulted in 3,918 leaf nodes. The parameters of the reading-style speech synthesizer for the speaker YMK were adapted to the utterances of the speaker SJK based on the speaker adaptation method supported by HTS [13]. The adapted parameters then constructed the reading-style speech synthesizer for the speaker SJK.

To build a singing voice database, we collected 95 songs amounting to 105 minutes sung by the speaker SJK. Spectrum features of the singing voice data were extracted and represented in the same manner with those of the reading-style utterances. In conjunction with the recorded sounds, the musical score associated with each song was also provided. Since the pitch/duration-related information is available from the musical notes accompanied to the singing voice, it is necessary to adapt only the parameters related to the spectrum features.

When applying MRHSMM and FMLLR, the pitch and duration derived from each musical note were used as the control parameter, $\eta = (\tilde{P}, \tilde{D})$ where $\tilde{P}$ is the fundamental frequency of the note written in the unit of Hz and $\tilde{D}$ indicates the duration given as the number of frames. As for the control vector, we set ξ_t as

$$\xi_t = (1, log\tilde{P}(t), log\tilde{D}(t))'. \tag{12}$$

From a number of preliminary experiments, we could find that the control vector as given by (12) produced stable speech quality while guaranteeing robust parameter estimation.

3.1. Objective performance evaluation for singing voice synthesis

Four different methods were compared in the experiments. We tried MLLR with a diagonal structure of the regression matrix $\mathbf{M}$ (MLLR_d) [11]. In the second method denoted as MR_H, the HMM mean vectors were expressed by the multiple regression matrix $\mathbf{H}_s$ as given in [8] with the corresponding control vector (12). The duration models in the MRHSMM method were ignored in the experiment since it was given as a specific length of the note. The third and last methods are denoted by FMLLR_d and FMLLR_f in which the HMM mean vectors were transformed differently via the proposed diagonal and full matrix FMLLR techniques, respectively.

Among 95 songs provided by the speaker SJK, we used 80 songs for training the regression matrices of each method and the remaining 15 songs for evaluating performances. Considering the size of the decision tree for the reading-style speech synthesizer, the amount of the singing voice data is considered quite small to expect a good performance of adaptation. To alleviate this difficulty, we applied a tying approach to the estimation of the regression matrices. Fig. 1 shows the average cepstral distance, i.e., squared difference between the mel-scaled cep-

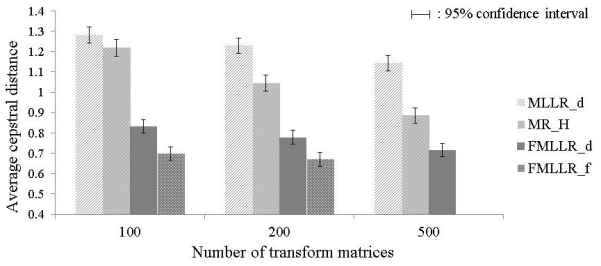

Figure 1: Average cepstral distance between the original and synthesized singing voice.

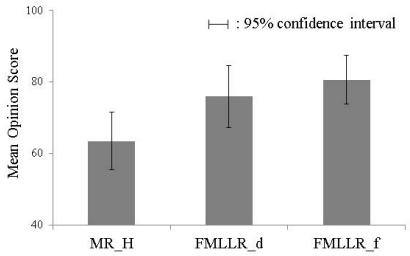

Figure 2: Results of subjective quality test for singing voice.

stra achieved from the original and synthesized singing voices obtained by varying the number of regression matrices. From the results we can find that the FMLLR approaches much reduced the cepstral discrepancy than other techniques. Note that in the case of FMLLR_f, we could increase the number of regression matrices only up to 200 due to the inadequate training data relative to the number of parameters to be estimated. We can also see that FMLLR_f produced the least cepstral distance even with 200 regression matrices compared with other methods. For FMLLR_d, it showed similar performance to that of FMLLR_f when the number of regression matrices exceeded 500.

3.2. Subjective listening test for singing voice synthesizer

We performed a subjective listening test for which 10 listeners participated. In the test, each listener was provided with singing voices synthesized from different methods, and gave his/her opinion for speech quality as a score in the range of [0, 100], a low score for poor singing quality and a high score for good quality.

The method of signal generation in the singing voice synthesizer is almost same to that of the reading-style speech synthesizer except that the lyrics are synchronized with the musical notes which control the pitch and duration [14]. Unlike [14], we applied the values of the note such as the pitch and duration to the HMM directly without any statistical modeling. Therefore the quality of the synthesized singing voice mostly depends on the spectrums which are computed from the adapted HMM parameters.

Three methods, MR_H, FMLLR_d and FMLLR_f, were applied to generate the singing voice of 8 songs which were not included in the training database. Control parameter for the MRHSMM and FMLLR methods was determined as (12). The number of regression matrices for each method was set to 200. From the obtained scores shown in Fig. 2, we can see that the proposed FMLLR approaches produced a better quality than the other approaches to singing voice synthesis. Moreover, FMLLR_f showed a higher subjective quality score than FMLLR_d.

4. Conclusions

In this paper, we have extended the structure of FMLLR from diagonal to full matrix which depends on varying control parameters and presented a training procedure based on the EM algorithm. To evaluate the performance of the proposed algorithm, we conducted several experiments with various conditions for synthesizing singing voice. It has been shown that the proposed algorithm is effective for the singing voice synthesizer to generate more natural singing voice quality.

5. Acknowledgements

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0083044) and by the Advanced Industrial Technology Development Program funded by the Ministry of Knowledge Economy (No. 10031489).

6. References

[1] C. J. Leggetter, and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171-185, Apr. 1995.

[2] M. Gales, "Cluster adaptive training of hidden Markov Models," *IEEE Trans. on speech and audio proc.*, vol. 8, no. 4, pp. 417-428, Jul. 2000.

[3] K. Visweswariah, V. Goel, and R. Gopinath, "Structured linear transforms for adaptation using training time information," *Proc. ICASSP*, pp. 585-288, 2002.

[4] B. Mak, and R. Hsiao, "Kernel eigenspace-based MLLR adaptation," *IEEE Trans. Audio, Speech and Language Process.*, vol. 15, no. 3, pp. 784-795, Mar. 2007.

[5] Z. Karam, and W. Campbell, "A multi-class MLLR kernel for SVM speaker recognition," in *Proc. ICASSP*, Las Vegas, NV, pp. 4117-4120, 2008.

[6] Y. Sung, C. Boulis, and D. Jurafsky, "Maximum conditional likelihood linear regression and maximum a posteriori for hidden conditional random fields speaker adaptation," in *Proc. ICASSP*, Las Vegas, NV, pp. 4293-4296, 2008.

[7] T. Nose, Y. Kato, and T. Kobayashi, "A speaker adaptation technique for MRHSMM-based style control of synthetic speech," in *Proc. ICASSP*, Honolulu, HI, pp. 833-836, 2007.

[8] M. Tachibana, S. Izawa, T. Nose, and T. Kobayashi, "Speaker and style adaptation using average voice model for style control in HMM-based speech synthesis," in *Proc. ICASSP*, Las Vegas, NE, pp. 4633-4636, 2008.

[9] T. Nose, M. Tachibana, and T. Kobayashi, "HMM-based style control for expressive speech synthesis with arbitrary speaker's voice using model adaptation," *IEICE Trans. Inf. and Syst.*, vol. E92-D, 3, pp. 489-497, Mar. 2009.

[10] J. Sundberg, "The acoustics of the singing voice," *Sci. Amer.*, pp. 82-91, Mar. 1977.

[11] N. S. Kim, J. S. Sung, and D. H. Hong, "Factored MLLR adaptation," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 99-102, Feb. 2011.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY, Springer, 2006.

[13] H. Zen et al., "The HMM-based speech synthesis system version 2.0," *Proc. of ISCA SSW6*, Bonn, Germany, Aug. 2007.

[14] K. Saino et al., "HMM-based singing voice synthesis system," *Proc. Interspeech*, pp. 1141-1144, Sep. 2006.