Speaker Adaptation of Trajectory HMMs Using Feature-Space MLLR

Heiga Zen, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura

Department of Computer Science and Engineering
Nagoya Institute of Technology, Nagoya, Japan
{zen,nankaku,tokuda,kitamura}@ics.nitech.ac.jp

Abstract

Recently, a trajectory model, derived from the hidden Markov model (HMM) by imposing explicit relationships between static and dynamic features, has been proposed. The derived model, named *trajectory HMM*, can alleviate two limitations of the HMM: constant statistics within a state and conditional independence assumption of state output probabilities. In the present paper, a speaker adaptation algorithm for the trajectory HMM based on feature-space Maximum Likelihood Linear Regression (fMLLR) is derived and evaluated. Results of a simple continuous speech recognition experiment shows that adapting trajectory HMMs using the derived adaptation algorithm improves the speech recognition performance.
Index Terms: trajectory HMM, adaptation, fMLLR.

1. Introduction

Speech recognition technologies have achieved significant progress with the introduction of hidden Markov models (HMMs). Their tractability and efficient implementations are achieved by a number of assumptions, such as constant statistics within an HMM state, conditional independence of state output probabilities. Although these assumptions make the HMM practically useful, they are not realistic for modeling sequences of speech spectra, especially in spontaneous speech. To overcome these shortcomings of the HMM, a variety of alternative models have been proposed, e.g., [1–3]. Although these models can improve the speech recognition performance, they generally require an increase in the number of model parameters and computational complexity. Alternatively, the use of dynamic features (delta and delta-delta features) [4] also improves the performance of HMM-based speech recognizers. It can be viewed as a simple mechanism to capture time dependencies. However, it has been thought of as an ad hoc rather than an essential solution. Generally, dynamic features are calculated as regression coefficients from their neighboring static features. Therefore, relationships between static and dynamic feature vector sequences are deterministic. However, usually these relationships are ignored and the static and dynamic features are modeled as independent random variables. Ignoring these dependencies allows inconsistency between the static and dynamic features when the HMM is used as a generative model in the obvious way.

Recently, a trajectory model, derived from the HMM by imposing the explicit relationships between static and dynamic features, has been proposed [5]. The derived model, named *trajectory HMM*, can overcome the above two limitations of the HMM without any additional parameters. Maximum likelihood (ML) training algorithms for the trajectory HMM based on the Viterbi and Monte Carlo approximations have also been derived [5, 6]. It was applied to speaker-dependent acoustic modeling not only in the speech recognition but also in the speech synthesis [5].

Currently, most of state-of-the-art speech recognition systems adopt speaker adaptation techniques. These techniques aim to adapt speaker-independent acoustic models to specific speakers to improve the speech recognition performance. In addition, these techniques are also used in the HMM-based speech synthesis framework [7] to construct a speaker-specific synthesis system using only a small amount of speech [8]. Generally, the speaker-adaptation techniques can roughly be clustered into three classes [9]; Maximum A Posteriori (MAP) adaptation [10], linear transformation based technique such as Maximum Likelihood Linear Regression (MLLR) [11], or speaker clustering/speaker space methods such as eigenvoice [12].

In the present paper, a speaker adaptation algorithm based on feature-space MLLR (fMLLR) [11] [1] for the trajectory HMM is derived and evaluated. Although the trajectory HMM has the same parameterization as the HMM, the definition of its output probability is different from that of the HMM. Accordingly, the adaptation algorithm should be re-derived based on its output probability.

The rest of the present paper is organized as follows: Section 2 reviews the definition of the trajectory HMM. In Section 3, fMLLR-based adaptation algorithm for the trajectory HMM is derived. Results of a continuous speech recognition experiment are shown in Section 4. Concluding remarks and future plans are presented in the final section.

[1] In the HMM, feature-space MLLR and constrained model-space MLLR are identical [11]. However, in the trajectory HMM case model space has higher dimensionality (typically three times) than that of feature space. Because of this property, constrained model-space MLLR and feature-space MLLR have different meaning. In the present paper, we consider the feature space transformation.

2. Definition of Trajectory HMMs

The output probability of an acoustic static feature vector sequence $\boldsymbol{c} = \left[\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top$ for a trajectory HMM Λ is given by

$$p(\boldsymbol{c} \mid \Lambda) = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda)\ P(\boldsymbol{q} \mid \Lambda), \tag{1}$$

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) = \mathcal{N}\left(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{2}$$

$$P(\boldsymbol{q} \mid \Lambda) = P(q_1 \mid \Lambda) \prod_{t=2}^{T} P(q_t \mid q_{t-1}, \Lambda), \tag{3}$$

where $\boldsymbol{c}_t$ is an M-dimensional acoustic static feature vector at time t (e.g., MFCC, PLP, etc.), $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is a state sequence,[2] q_t is the state at time t, and T is the number of frames in $\boldsymbol{c}$. In Eq. (2), $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ are the $MT \times 1$ mean vector (smooth trajectory) and the $MT \times MT$ temporal covariance matrix for $\boldsymbol{q}$, respectively. They are given by

$$\boldsymbol{R}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{r}_{\boldsymbol{q}}, \tag{4}$$

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}, \tag{5}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}, \tag{6}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top\right]^\top, \tag{7}$$

$$\boldsymbol{\mu}_{q_t} = \left[\Delta^{(0)}\boldsymbol{\mu}_{q_t}^\top, \Delta^{(1)}\boldsymbol{\mu}_{q_t}^\top, \Delta^{(2)}\boldsymbol{\mu}_{q_t}^\top\right]^\top, \tag{8}$$

$$\Delta^{(d)}\boldsymbol{\mu}_{q_t} = \left[\Delta^{(d)}\mu_{q_t}(1), \ldots, \Delta^{(d)}\mu_{q_t}(M)\right]^\top, \quad d = 0, 1, 2 \tag{9}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \mathrm{diag}\left[\boldsymbol{\Sigma}_{q_1}, \ldots, \boldsymbol{\Sigma}_{q_T}\right], \tag{10}$$

$$\boldsymbol{\Sigma}_{q_t} = \mathrm{diag}\left[\Delta^{(0)}\boldsymbol{\Sigma}_{q_t}, \Delta^{(1)}\boldsymbol{\Sigma}_{q_t}, \Delta^{(2)}\boldsymbol{\Sigma}_{q_t}\right], \tag{11}$$

$$\Delta^{(d)}\boldsymbol{\Sigma}_{q_t} = \mathrm{diag}\left[\Delta^{(d)}\sigma_{q_t}(1), \ldots, \Delta^{(d)}\sigma_{q_t}(M)\right], \quad d = 0, 1, 2 \tag{12}$$

where $\boldsymbol{\mu}_{q_t}$ and $\boldsymbol{\Sigma}_{q_t}$ are the $3M \times 1$ mean vector and the $3M \times 3M$ covariance matrix associated with the q_t-th state, respectively. In Eqs. (5) and (6), $\boldsymbol{W}$ is a $3MT \times MT$ window matrix whose elements are given as regression window coefficients to calculate delta and delta-delta as follows:

$$\Delta^{(1)}\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \quad \Delta^{(2)}\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{13}$$

[2]For notation simplicity, we assume that each state has a Gaussian density function with a diagonal covariance matrix for its output distribution.

$$\boldsymbol{W} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_T]^\top \otimes \boldsymbol{I}_{M\times M}, \tag{14}$$

$$\boldsymbol{W}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], \tag{15}$$

$$\boldsymbol{w}_t^{(d)} = \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(d)}-1}, w^{(d)}(-L_-^{(d)}), \ldots, w^{(d)}(0), \ldots, w^{(d)}(L_+^{(d)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(d)}\right)}\Big]^\top, d = 0, 1, 2 \tag{16}$$

$L_-^{(0)} = L_+^{(0)} = 0$, and $w^{(0)}(0) = 1$.

Note that $\boldsymbol{c}$ is modeled by a mixture of Gaussian density functions whose dimensionality is MT, and the covariance matrices of these Gaussian density functions are generally full. As a result, the trajectory HMM can alleviate the deficiencies of the HMM. It is also noted that the parameterization of the trajectory HMM is completely the same as that of the HMM with the same model topology.

3. Adaptation of Trajectory HMMs

Currently, many state-of-the-art speech recognition systems adopt speaker adaptation techniques. These techniques aim to adapt speaker-independent acoustic models to specific speakers to improve the speech recognition performance. One of the most popular speaker adaptation techniques is Maximum Likelihood Linear Regression (MLLR) [11] There are two well-known forms in the MLLR framework: model-space and feature-space (MLLR). In this section, the feature-space MLLR (fMLLR) algorithm is derived for the trajectory HMM.

In fMLLR for the trajectory HMM, an acoustic static feature vector sequence $\boldsymbol{c}$ is transformed using an $MT \times MT$ linear transformation matrix $\boldsymbol{A}$ and an $MT \times 1$ bias vector $\boldsymbol{b}$ as follows:

$$\hat{\boldsymbol{c}} = \left[\hat{\boldsymbol{c}}_1^\top, \ldots, \hat{\boldsymbol{c}}_T^\top\right]^\top = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{b}. \tag{17}$$

Thus, the output probability of $\boldsymbol{c}$ conditioned on $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{q}$ is given by

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \boldsymbol{A}, \boldsymbol{b}, \Lambda) = |\boldsymbol{A}|\, \mathcal{N}\left(\hat{\boldsymbol{c}} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) \tag{18}$$

$$= \mathcal{N}\left(\boldsymbol{c} \mid \boldsymbol{A}^{-1}(\bar{\boldsymbol{c}}_{\boldsymbol{q}} - \boldsymbol{b}), \boldsymbol{A}^{-1}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{A}^{-\top}\right) \tag{19}$$

The goal of fMLLR for the trajectory HMM is to find $\boldsymbol{A}$ and $\boldsymbol{b}$ which maximize the model likelihood for given adaptation data $\boldsymbol{c}$.

In common with fMLLR for the HMM, the expectation-maximization (EM) algorithm can be used. The auxiliary function of the EM algorithm is defined as

$$Q(\Lambda, \Lambda') = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{q} \mid \boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b}, \Lambda) \cdot \Big[K + \log|\boldsymbol{A}| - \frac{1}{2}\left\{\left(\boldsymbol{A}\boldsymbol{c} + \boldsymbol{b} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)^\top \boldsymbol{P}_{\boldsymbol{q}}^{-1}\left(\boldsymbol{A}\boldsymbol{c} + \boldsymbol{b} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)\right\}\Big], \tag{20}$$

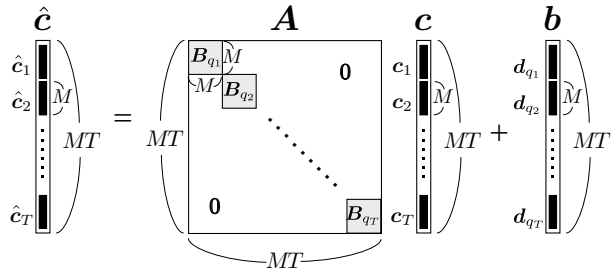

Figure 1: *Constraints of a linear transformation matrix* $\boldsymbol{A}$ *and a bias vector* $\boldsymbol{b}$ *introduced in fMLLR for the trajectory HMM.*

where K is a constant independent of $\boldsymbol{A}$ and $\boldsymbol{b}$. Although $\boldsymbol{A}$ and $\boldsymbol{b}$ could be estimated using Eq. (20), the total number of parameters is $MT \times (MT + 1)$: it is difficult to estimate statistically reliable $\boldsymbol{A}$ and $\boldsymbol{b}$ using a limited amount of adaptation data because the number of parameters to be estimated is huge (If $T = 1000$ and $M = 13$, total number of parameters of the linear transform is 1.82×10^8). To avoid this problem, in the present paper we introduce constraints into the structure of $\boldsymbol{A}$ and $\boldsymbol{b}$ (see also Fig. 1): $\boldsymbol{A}$ is block-diagonal and $\boldsymbol{A}$ and $\boldsymbol{b}$ have the parameter sharing structure which depends on $\boldsymbol{q}$. For notation simplicity, we assume that single linear transform is shared over all Gaussian density functions in a model set.[3] Under this assumption, the transformation matrix $\boldsymbol{A}$ and bias vector $\boldsymbol{b}$ can be written as

$$\boldsymbol{A} = \mathrm{diag}\Big[\underbrace{\boldsymbol{B}, \ldots, \boldsymbol{B}}_{T}\Big] \qquad \boldsymbol{b} = \Big[\underbrace{\boldsymbol{d}^\top, \ldots, \boldsymbol{d}^\top}_{T}\Big]^\top, \tag{21}$$

where $\boldsymbol{B}$ and $\boldsymbol{d}$ are an $M \times M$ transformation matrix and an $M \times 1$ bias vector shared over all Gaussian density functions in the model set, respectively. As a result, $\hat{\boldsymbol{c}}_t$ can be written using $\boldsymbol{c}_t$, $\boldsymbol{B}$ and $\boldsymbol{d}$ as follows:

$$\hat{\boldsymbol{c}}_t = \boldsymbol{B}\boldsymbol{c}_t + \boldsymbol{d} = \boldsymbol{X}\boldsymbol{\xi}_t, \tag{22}$$

where $\boldsymbol{X} = [\boldsymbol{d}^\top, \boldsymbol{B}^\top]^\top$ is an $(M + 1) \times M$ extended transformation matrix, $\boldsymbol{\xi}_t = [1, \boldsymbol{c}_t^\top]^\top$ is an $(M + 1) \times 1$ extended acoustic static feature vector at time t.

Under the above assumptions, Eq. (20) can be reformulated as the same manner in [11]:

$$Q(\Lambda, \Lambda') = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{q} \mid \boldsymbol{c}, \boldsymbol{X}, \Lambda) \cdot \left[K + T \log \left| \boldsymbol{p}_m \boldsymbol{x}_m^\top \right| - \frac{1}{2} \sum_{m=1}^{M} \left\{ \boldsymbol{x}_m \boldsymbol{G}_{\boldsymbol{q}}^{(m)} \boldsymbol{x}_m^\top - 2 \boldsymbol{x}_m \boldsymbol{k}_{\boldsymbol{q}}^{(m)\top} \right\} \right] \tag{23}$$

[3] If the number of transforms is larger than 1, all transforms are depend on each other. Therefore, optimization should be iterated not only over rows but also transforms.

where

$$\boldsymbol{\zeta}_t^{(d)} = \sum_{\tau=-L_-^{(d)}}^{L_+^{(d)}} w^{(d)}(\tau) \cdot \boldsymbol{\xi}_{t+\tau} \tag{24}$$

$$\boldsymbol{G}_{\boldsymbol{q}}^{(m)} = \sum_{t=1}^{T} \sum_{d=0}^{2} \frac{1}{\Delta^{(d)} \sigma_{q_t}(m)} \boldsymbol{\zeta}_t^{(d)} \cdot \boldsymbol{\zeta}_t^{(d)\top} \tag{25}$$

$$\boldsymbol{k}_{\boldsymbol{q}}^{(m)} = \sum_{t=1}^{T} \sum_{d=0}^{2} \frac{1}{\Delta^{(d)} \sigma_{q_t}(m)} \Delta^{(d)} \mu_{q_t}(m) \cdot \boldsymbol{\zeta}_t^{(d)}. \tag{26}$$

In Eq. (23), $\boldsymbol{x}_m$ corresponds to the m-th row of $\boldsymbol{X}$ and $\boldsymbol{p}_m$ is an extended cofactor row vector given by

$$\boldsymbol{p}_m = [0, c_{(m,1)}, \ldots, c_{(m,M)}] \tag{27}$$

$$c_{(m,n)} = \mathrm{cof}\left(\boldsymbol{B}_{m,n}\right). \tag{28}$$

Taking the partial derivative of Eq. (23) with respect to $\boldsymbol{x}_m$ yields

$$\frac{\partial Q(\Lambda, \Lambda')}{\partial \boldsymbol{x}_m} = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{q} \mid \boldsymbol{c}, \boldsymbol{X}, \Lambda) \cdot \left\{ \frac{\boldsymbol{p}_m}{\boldsymbol{p}_m \boldsymbol{x}_m^\top} - \boldsymbol{x}_m \boldsymbol{G}_{\boldsymbol{q}}^{(m)} + \boldsymbol{k}_{\boldsymbol{q}}^{(m)} \right\} \tag{29}$$

Using the above equation, $\boldsymbol{X}$ can be iteratively updated using a row by row optimization technique described in [11].

For exact computation of Eq. (29), all possible state sequences should be evaluated. However, it is intractable because the temporal covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$ of the trajectory HMM is generally full. Therefore, approximations such as Viterbi approximation [5] or Markov Chain Monte Carlo [6] should be introduced.

4. Experiments

4.1. Experimental conditions

Phonetically balanced 440 of 503 sentences uttered by male speakers MHO, MMY, MSH, MTK, and MYI (2200 sentences in total) from the ATR Japanese speech database B-set were used for training context-independent HMMs and trajectory HMMs. Remaining 10 and 53 sentences uttered by a male speaker MHT were used for adaptation and evaluation, respectively. These test utterances had an average length of 43 phonemes and an average duration of 4 seconds.

Speech signals were sampled at 16 kHz and windowed by a 25-ms Blackman window with a 10-ms shift, and then mel-cepstral coefficients were obtained by a mel-cepstral analysis technique. Static feature vectors consisted of 19 mel-cepstral coefficients including the zeroth coefficient. They were augmented by appending their first and second order dynamic features.

The three-state left-to-right with no-skip structure was used for modeling 36 Japanese phonemes including silence and short pause. Each state had a single Gaussian density function with a diagonal covariance matrix. After training

Table 1: *Phoneme Error Rates (PER) of the HMMs and the trajectory HMMs with and without speaker adaptation (1000+1000 best lists rescoring).*

Model	Adaptation	PER (%)
HMM	w/o adapt.	49.3
	with adapt. (EM)	34.2
trajectory HMM	w/o adapt.	49.6
	with adapt. (Viterbi)	32.7
	with adapt. (MCEM)	32.8

the HMMs in the standard way, the trajectory HMMs were iteratively reestimated (two iterations) by the Viterbi training [5] using the HMMs as its initial models. The number of delay of the delayed decision Viterbi algorithm [5] for searching better state sequences was four (beam width was 1500).

In this experiment, static supervised adaptation was used. For adapting the HMMs, a block-diagonal transformation matrix structure consisting of three 19×19 blocks was adopted. The trajectory HMMs were adapted using a 19×19 transformation matrix and a 19×1 bias vector. Therefore, the transform of the HMMs had three times larger number of parameters than that of the trajectory HMMs. For approximating Eq. (29), we used the Viterbi approximation or the Monte Carlo EM (MCEM) algorithm with 100 samples.

4.2. Experimental results

In the recognition experiment reported in this section, the rescoring paradigm was used. Two 1000-best list sets were generated for each test utterance by the HTK Viterbi decoder using the HMMs with and without speaker adaptation. These two 1000-best list sets were merged and then resegmented and rescored by the trajectory HMMs. To give an idea of the range of merged 1000+1000-best lists, the error rates of the best, worst, and average of randomly selected hypotheses (100 times) were 24.4%, 59.5%, and 42.7%, respectively. The best (24.4%) and worst (59.5%) error rates were the bounds on subsequent rescoring results.

Table 1 shows the phoneme error rates of the HMMs and trajectory HMMs with and without speaker adaptation. In the table, EM, Viterbi, and MCEM denotes that the linear transforms were estimated using the exact EM algorithm, the Viterbi approximation, and the MCEM algorithm with 100 samples, respectively. It shows that the speaker-adapted trajectory HMMs achieved 17% relative error reduction over the trajectory HMM without adaption. Although the recognition performance of the HMMs and the trajectory HMMs without adaptation was almost the same, the adapted trajectory HMMs achieved 4% relative error reduction over the HMMs with adaptation in this experiment. We could not see any significant difference in the recognition performance between Viterbi training and MCEM algorithm for estimating the linear transforms.

5. Conclusion

In the present paper, a speaker adaptation technique for the trajectory HMM based on feature-space MLLR was derived and evaluated. The speaker-adapted trajectory HMMs achieved 17% and 4% relative error reduction over the trajectory HMMs without adaptation and the corresponding HMMs with adaptation, respectively.

Future plan includes introducing the different constraints into the structure of the linear transforms and evaluating the performance with multiple transforms. Large-scale evaluation is also necessary.

6. Acknowledgement

This work was partly supported by MEXT e-Society project.

7. References

[1] M. Ostendorf, V. Digalakis, and O.A. Kimball, "From HMMs to segment models," *IEEE Transactions on Speech & Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.

[2] A.V.I. Rosti and M.J.F Gales, "Switching linear dynamical systems for speech recognition," Tech. Rep. CUED/F-INFENG/TR.461, Cambridge University, 2003.

[3] G. Zweig, *Speech recognition using dynamic Bayesian networks*, Ph.D. thesis, University of California, Berkeley, 1998.

[4] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions Acoustics, Speech, & Signal Processing*, vol. 34, pp. 52–59, 1986.

[5] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer Speech & Language*, 2006, (accepted).

[6] H. Zen, K. Tokuda, and T. Kitamura, "Estimating trajectory HMM parameters by Monte Carlo EM with Gibbs sampler," in *Proc. of ICASSP*, 2006, vol. I, pp. 1173–7356.

[7] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. of Eurospeech*, 1999, pp. 2347–2350.

[8] J. Yamagishi, *Average-Voice-Based Speech Synthesis*, Ph.D. thesis, Tokyo Institute of Technology, Japan, 2006.

[9] P.C. Woodland, "Speaker adaptation for continuous density hmms: A review," in *ITRW on adaptation methods for speech recognition*, 2001, pp. 11–19.

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

[11] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[12] R. Kuhn, J.C. Janqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000.