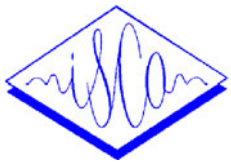

7th ISCA Workshop on
Speech Synthesis (SSW-7)
Kyoto, Japan
September 22-24, 2010

Linear Transformation Approaches to Many-to-One Voice Conversion

Chie Hayashida†, Tomoki Toda, Yamato Ohtani‡, Hiroshi Saruwatari, Kiyohiro Shikano

Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan

tomoki@is.naist.jp

Abstract

In this paper, we present linear transformation algorithms for many to one voice conversion (VC). Many to one VC is a tech nique for converting an arbitrary source speaker's voice into the target speaker's voice. A conversion model previously devel oped between many prestored source speakers and the target speaker is adapted into a new source speaker in an unsuper vised manner. In this study, we implement several well known model adaptation techniques based on linear transformation for many to one VC and evaluate their effectiveness.

Index Terms: many to one voice conversion, Gaussian mixture model, unsupervised adaptation, linear transformation

1. Introduction

Voice conversion (VC) has attracted attention as a potential technique for developing a new speech communication system capable of controlling the voice quality beyond physical con straints. Speaker conversion has been studied as one of the typ ical VC applications and various statistical approaches, such as a method based on a Gaussian mixture model (GMM) [1], have been proposed. The traditional VC framework typically uses a parallel data set consisting of utterance pairs of a source speaker and a target speaker to develop a conversion model between those speakers.

To relax the constraint that the parallel data set is required, a training framework accepting nonparallel data has been studied recently. Mouchtaris *et al.* [2] proposed a nonparallel training method based on maximum likelihood stochastic transforma tion (MLST) [3]. Lee and Wu [4] applied maximum *a poste riori* (MAP) adaptation [5] to a nonparallel training framework using a MAP based training method [6]. The MLST and MAP adaptation were originally developed for speech recognition and their effectiveness has been confirmed. It is promising to apply these techniques to the VC framework.

Inspired by the model adaptation techniques in speech recognition, many to one VC and one to many VC frameworks have been proposed [7], where an arbitrary source speaker's voice is converted into the target speaker's voice in many to one VC and vice versa in one to many VC. An eigenvoice technique [8] has successfully been applied to these frameworks [7], and it has further been extended to many to many VC [9]. One of the notable advantages of these frameworks is the effective use of many prestored speakers' voices for developing a conversion model for new speakers.

In many to one VC, since the amount of adaptation data continuously increases while the source speaker continues to use the VC system, adaptation techniques according to the amount of adaptation data are helpful. The eigenvoice tech nique is capable of rapidly adapting the conversion model even if only a short speech segment (approximately 300 ms in [10]) is available as the adaptation data; its adaptation performance is slightly improved even if the amount of adaptation data in creases. It is well known that adaptation techniques with a larger number of adaptation parameters, such as linear transfor mation, yield a higher adaptation performance than the eigen voice technique when the amount of adaptation data increases.

In this paper, we present linear transformation approaches to many to one VC. As successful model adaptation techniques, maximum likelihood linear regression (MLLR) [11] and con strained MLLR (CMLLR) [12, 13] are applied to the many to one VC framework. The effectiveness of the CMLLR adap tation in the GMM based VC framework has been confirmed under some limited conditions: *e.g.*, diagonal transforms are used for adapting the GMM for a single speaker pair [2] or the adaptation is performed between not different speakers but dif ferent recording conditions in body conducted speech enhance ment [14]. Therefore, it is still an open question whether it is effective in many to one VC. Moreover, speaker adaptive train ing (SAT) [15] and MAP estimation [5], which are well known techniques for improving the adaptation performance, are also applied to the many to one VC framework. The effectiveness of the proposed methods is shown in the experimental results.

2. Traditional VC

In this section, we describe a traditional VC framework based on a GMM of the joint probability density of the source and tar get acoustic feature vectors [1, 16]. In this study, we employ the maximum likelihood trajectory based conversion method [17].

2.1. Joint Probability Density Modeling with GMM

Let $\boldsymbol{X}_t = \left[\boldsymbol{x}_t^\top, \Delta\boldsymbol{x}_t^\top\right]^\top$ and $\boldsymbol{Y}_t = \left[\boldsymbol{y}_t^\top, \Delta\boldsymbol{y}_t^\top\right]^\top$ be $2D$ dimensional source and target acoustic feature vectors, each of which consists of D dimensional static and dynamic feature vectors at frame t, where $^\top$ denotes the transposition of a vec tor. The joint probability density function modeled by a GMM is given by

$$P(\boldsymbol{X}_t, \boldsymbol{Y}_t|\boldsymbol{\lambda}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix}\boldsymbol{X}_t \\ \boldsymbol{Y}_t\end{bmatrix}; \boldsymbol{\mu}_m^{(X,Y)}, \boldsymbol{\Sigma}_m^{(X,Y)}\right), \quad (1)$$

where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ shows the normal distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The total number of mix ture components is M. The weight of the m^{th} mixture compo nent is α_m. The parameter set $\boldsymbol{\lambda}$ consists of the weights, mean vectors, and covariance matrices of individual mixture compo nents. The mean vector and covariance matrix of the m^{th} mix ture component are written as

$$\boldsymbol{\mu}_m^{(X,Y)} = \begin{bmatrix}\boldsymbol{\mu}_m^{(X)} \\ \boldsymbol{\mu}_m^{(Y)}\end{bmatrix}, \quad \boldsymbol{\Sigma}_m^{(X,Y)} = \begin{bmatrix}\boldsymbol{\Sigma}_m^{(XX)} & \boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)}\end{bmatrix}, \quad (2)$$

where $\boldsymbol{\mu}_m^{(X)}$ and $\boldsymbol{\mu}_m^{(Y)}$ are the source and target mean vectors, respectively. The block matrices $\boldsymbol{\Sigma}_m^{(XX)}$ and $\boldsymbol{\Sigma}_m^{(YY)}$ are the source and target covariance matrices, respectively. The block matrices $\boldsymbol{\Sigma}_m^{(XY)}$ and $\boldsymbol{\Sigma}_m^{(YX)}$ are the cross covariance matrices between the source and target feature vectors.

†Presently, with NTT DATA CORPORATION, Japan.
‡Presently, with TOSHIBA CORPORATION, Japan.

2.2. Training and Conversion

A parallel data set consisting of the source and target speaker's voices is used for training the GMM. Using the time aligned source and target feature vectors generated from the parallel data set, the GMM parameter set is optimized in the sense of maximum likelihood with the EM algorithm.

In conversion, a time sequence of target static feature vec tors $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T\}$ for a time sequence of given source feature vectors $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_T\}$ is determined by maximizing a likeli hood function defined as a product of the conditional probabil ity density function $P(\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_T|\boldsymbol{X}_1, \cdots, \boldsymbol{X}_T, \boldsymbol{\lambda})$ derived from the trained GMM and the probability density function of the global variance (GV), which is the variance of the target static feature vectors calculated over a time sequence. This con version method effectively considers interframe correlation and a higher order moment for inhibiting discontinuities and over smoothing effects in the converted feature vectors.

3. Many-to-One VC

The many to one VC framework consists of 1) training, 2) adaptation, and 3) conversion processes. The training process employs multiple parallel data sets consisting of utterance pairs of many prestored source speakers and a single target speaker for developing the initial conversion model. The adaptation pro cess adapts the initial conversion model to a new source speaker using only his/her speech data without any linguistic restric tions. The conversion process is straightforwardly performed with the adapted conversion model. This framework does not require a parallel data set between the new source speaker and the target speaker.

Several many to one VC algorithms have been proposed [7, 10]. The source speaker independent model (SI GMM) is applied to many to one VC without any adaptation processes. The use of model adaptation techniques, such as speaker selec tion and eigenvoice techniques, yields significant improvements in conversion performance.

4. Many-to-One VC Algorithms Based on Constrained Linear Transformation

In this section, linear transformation techniques based on the CMLLR adaptation are applied to many to one VC.

4.1. CMLLR for Many-to-One VC

The CMLLR adaptation for many to one VC linearly trans forms the source feature vector as

$$\tilde{\boldsymbol{X}}_t = \boldsymbol{A}\boldsymbol{X}_t + \boldsymbol{b} = \boldsymbol{W}\boldsymbol{\zeta}_t, \tag{3}$$

where $\boldsymbol{W}$ is the extended transform $[\boldsymbol{b}, \boldsymbol{A}]$ and $\boldsymbol{\zeta}_t$ is the ex tended source feature vector $[1, \boldsymbol{X}_t^\top]^\top$ at frame t. This feature space transformation is equivalent to the constrained model space transformation given by

$$P(\boldsymbol{X}_t, \boldsymbol{Y}_t|\boldsymbol{\lambda}, \boldsymbol{W}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix}\boldsymbol{X}_t \\ \boldsymbol{Y}_t\end{bmatrix}; \tilde{\boldsymbol{\mu}}_m^{(X,Y)}, \tilde{\boldsymbol{\Sigma}}_m^{(X,Y)}\right) \tag{4}$$

$$\tilde{\boldsymbol{\mu}}_m^{(X,Y)} = \begin{bmatrix}\tilde{\boldsymbol{\mu}}_m^{(X)} \\ \boldsymbol{\mu}_m^{(Y)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}'\boldsymbol{\mu}_m^{(X)} + \boldsymbol{b}' \\ \boldsymbol{\mu}_m^{(Y)}\end{bmatrix} \tag{5}$$

$$\tilde{\boldsymbol{\Sigma}}_m^{(X,Y)} = \begin{bmatrix}\tilde{\boldsymbol{\Sigma}}_m^{(XX)} & \tilde{\boldsymbol{\Sigma}}_m^{(XY)} \\ \tilde{\boldsymbol{\Sigma}}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}'\boldsymbol{\Sigma}_m^{(XX)}\boldsymbol{A}'^\top & \boldsymbol{A}'\boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{A}'^\top & \boldsymbol{\Sigma}_m^{(YY)}\end{bmatrix}, \tag{6}$$

where $\boldsymbol{A}' = \boldsymbol{A}^{-1}$ and $\boldsymbol{b}' = -\boldsymbol{A}^{-1}\boldsymbol{b}$.

4.2. Unsupervised Adaptation

In many to one VC, the CMLLR adaptation is performed using only the adaptation data of a new source speaker. For a time sequence of given source feature vectors $\left\{\boldsymbol{X}_1^{(new)}, \cdots, \boldsymbol{X}_T^{(new)}\right\}$, the extended transform is estimated by maximizing the total likelihood of the GMM of the marginal probability density function of the source feature vectors as

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} \prod_{t=1}^{T} P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \boldsymbol{W}). \tag{7}$$

The auxiliary function maximized in the EM algorithm is given by

$$\mathcal{Q}^{(ML)}(\boldsymbol{W}, \hat{\boldsymbol{W}}) = \sum_{t=1}^{T} \gamma_{m,t}^{(new)} \log P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \hat{\boldsymbol{W}}) \tag{8}$$

$$\gamma_{m,t}^{(new)} = P(m|\boldsymbol{X}_t^{(new)}, \boldsymbol{\lambda}, \boldsymbol{W}). \tag{9}$$

In the E step, the sufficient statistics are calculated as

$$\gamma_m^{(new)} = \sum_{t=1}^{T} \gamma_{m,t}^{(new)} \tag{10}$$

$$\langle\boldsymbol{\zeta}\rangle_m^{(new)} = \sum_{t=1}^{T} \gamma_{m,t}^{(new)} \boldsymbol{\zeta}_t \tag{11}$$

$$\langle\boldsymbol{\zeta}\boldsymbol{\zeta}^\top\rangle_m^{(new)} = \sum_{t=1}^{T} \gamma_{m,t}^{(new)} \boldsymbol{\zeta}_t \boldsymbol{\zeta}_t^\top. \tag{12}$$

In the M step, the extended transform is updated by row by row optimization [13, 18]. The ML estimate of the i^{th} row vector $\boldsymbol{w}_{(i)}$ of the extended transform $\boldsymbol{W}$ is given by

$$\hat{\boldsymbol{w}}_{(i)} = (\rho\boldsymbol{c}_i + \boldsymbol{k}_i)\boldsymbol{G}_{i,i}^{-1} \tag{13}$$

$$\boldsymbol{G}_{i,j} = \sum_{m=1}^{M} p_{m,(i,j)}^{(XX)} \langle\boldsymbol{\zeta}\boldsymbol{\zeta}^\top\rangle_m^{(new)} \tag{14}$$

$$\boldsymbol{k}_i = \sum_{m=1}^{M} \boldsymbol{p}_{m,(i)}^{(XX)} \boldsymbol{\mu}_m^{(X)} \langle\boldsymbol{\zeta}\rangle_m^{(new)^\top} - \sum_{j=1, j\neq i}^{2D} \boldsymbol{w}_{(j)} \boldsymbol{G}_{i,j}. \tag{15}$$

The vector $\boldsymbol{c}_i$ is the i^{th} extended cofactor row vector $[0, \text{cof}(\boldsymbol{A})_{i,1}, \cdots, \text{cof}(\boldsymbol{A})_{i,2D}]$, where $\text{cof}(\boldsymbol{A})_{i,j}$ is the cofac tor of the $(i,j)^{\text{th}}$ element of $\boldsymbol{A}$. The vector $\boldsymbol{p}_{m,(i)}^{(XX)}$ and the value $p_{m,(i,j)}^{(XX)}$ are the i^{th} row vector and the $(i,j)^{\text{th}}$ element of the precision matrix $\boldsymbol{\Sigma}_m^{(XX)^{-1}}$, respectively. The value ρ is determined by solving a quadratic equation, the constant term of which consists of $\gamma_m^{(new)}$, as described in [13]. This update should be performed iteratively over the rows because the ML estimate of each row vector depends on the other row vectors due to the cofactors. Note that the second term in the R.H.S. of Eq. (15) disappears when the diagonal matrix is used as every source covariance matrix $\boldsymbol{\Sigma}_m^{(XX)}$.

It is possible to use multiple transforms in the CMLLR adaptation. A regression tree is previously developed so as to hierarchically cluster the mixture components. In adaptation, the regression classes, the amount of adaptation data of which is larger than the minimum occupancy count, are determined dynamically [19]. The transform for each regression class is es timated with only parameters related to the corresponding mix ture components: *i.e.*, the summations in Eqs. (14) and (15) are

performed over the mixture components clustered into the same regression class.

4.3. Implementation of SAT

There are two main approaches to developing the GMM to which the linear transform is applied in adaptation. One is to use the SI GMM [7]. The other is to use a *canonical* GMM given in the SAT paradigm. In SAT, both the canonical GMM parameters and a set of speaker dependent linear transforms are optimized on the basis of a single objective function.

In SAT for many to one VC based on CMLLR, the canonical GMM parameter set $\boldsymbol{\lambda}$ and a set of prestored source speaker dependent linear transforms $\boldsymbol{W}^{(1:S)} = \left\{\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(S)}\right\}$ are optimized as

$$\left\{\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}^{(1:S)}\right\} = \underset{\left\{\boldsymbol{\lambda}, \boldsymbol{W}^{(1:S)}\right\}}{\operatorname{argmax}} \prod_{s=1}^{S}\prod_{t=1}^{T_s} P(\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t|\boldsymbol{\lambda}, \boldsymbol{W}^{(s)}), \quad (16)$$

where $\boldsymbol{X}_t^{(s)}$ and $\boldsymbol{Y}_t$ are the s^{th} prestored source speaker's feature vector and the target feature vector at frame t, respectively, in their time aligned parallel data set. The EM algorithm is used for maximizing the auxiliary function given by

$$\mathcal{Q}^{(SAT)}\left(\left\{\boldsymbol{\lambda}, \boldsymbol{W}^{(1:S)}\right\}, \left\{\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}^{(1:S)}\right\}\right) = \sum_{t=1}^{T} \gamma_{m,t}^{(s)} \log P(\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t|\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}^{(s)}) \quad (17)$$

$$\gamma_{m,t}^{(s)} = P(m|\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t, \boldsymbol{\lambda}, \boldsymbol{W}^{(s)}). \quad (18)$$

In the E step, the sufficient statistics are calculated as

$$\gamma_m^{(s)} = \sum_{t=1}^{T} \gamma_{m,t}^{(s)} \quad (19)$$

$$\begin{bmatrix} \langle\boldsymbol{\zeta}\rangle_m^{(s)} \\ \langle\boldsymbol{Y}\rangle_m^{(s)} \end{bmatrix} = \sum_{t=1}^{T} \gamma_{m,t}^{(s)} \begin{bmatrix} \boldsymbol{\zeta}_t^{(s)} \\ \boldsymbol{Y}_t \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \langle\boldsymbol{\zeta}\boldsymbol{\zeta}^\top\rangle_m^{(s)} & \langle\boldsymbol{\zeta}\boldsymbol{Y}^\top\rangle_m^{(s)} \\ \langle\boldsymbol{Y}\boldsymbol{\zeta}^\top\rangle_m^{(s)} & \langle\boldsymbol{Y}\boldsymbol{Y}^\top\rangle_m^{(s)} \end{bmatrix} = \sum_{t=1}^{T} \gamma_{m,t}^{(s)} \begin{bmatrix} \boldsymbol{\zeta}_t^{(s)} \\ \boldsymbol{Y}_t \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_t^{(s)} \\ \boldsymbol{Y}_t \end{bmatrix}^\top. \quad (21)$$

In the M step, the extended transforms for individual prestored source speakers and the canonical GMM parameter set are updated sequentially. First, the extended transform for each prestored source speaker is updated by row by row optimization. The i^{th} row vector $\boldsymbol{w}_{(i)}^{(s)}$ of the extended transform $\boldsymbol{W}^{(s)}$ for the s^{th} prestored source speaker is updated as

$$\hat{\boldsymbol{w}}_{(i)}^{(s)} = (\rho\boldsymbol{c}_i^{(s)} + \boldsymbol{k}_i^{(s)})\boldsymbol{G}_{i,i}^{(s)^{-1}} \quad (22)$$

$$\boldsymbol{G}_{i,j}^{(s)} = \sum_{m=1}^{M} p_{m,(i,j)}^{(X,Y)} \langle\boldsymbol{\zeta}\boldsymbol{\zeta}^\top\rangle_m^{(s)} \quad (23)$$

$$\boldsymbol{k}_i^{(s)} = \sum_{m=1}^{M} \boldsymbol{p}_{m,(i)}^{(X,Y)} \boldsymbol{\mu}_m^{(X,Y)} \langle\boldsymbol{\zeta}\rangle_m^{(s)^\top} - \sum_{j=1, j\neq i}^{2D} \boldsymbol{w}_{(j)}^{(s)} \boldsymbol{G}_{i,j}^{(s)} - \sum_{j=1}^{2D}\sum_{m=1}^{M} p_{m,(i,j+2D)}^{(X,Y)} \langle y\boldsymbol{\zeta}^\top\rangle_{m,(j)}^{(s)}. \quad (24)$$

The vector $\boldsymbol{c}_i^{(s)}$ is the i^{th} extended cofactor row vector of $\boldsymbol{A}^{(s)}$. The vector $\boldsymbol{p}_{m,(i)}^{(X,Y)}$ and the value $p_{m,(i,j)}^{(X,Y)}$ are the i^{th} row vector and the $(i,j)^{\text{th}}$ element of the precision matrix $\boldsymbol{\Sigma}_m^{(X,Y)^{-1}}$, respectively. The vector $\langle y\boldsymbol{\zeta}^\top\rangle_{m,(j)}^{(s)}$ is the j^{th} row vector of the matrix $\langle\boldsymbol{Y}\boldsymbol{\zeta}^\top\rangle_m^{(s)}$. This update process is regarded as supervised CMLLR adaptation because the joint feature vectors of the source and target speakers are used as the adaptation data. Then, the canonical GMM parameters are updated as

$$\hat{\alpha}_m = \frac{1}{\sum_{m=1}^{M}\sum_{s=1}^{S}\gamma_m^{(s)}} \sum_{s=1}^{S} \gamma_m^{(s)} \quad (25)$$

$$\hat{\boldsymbol{\mu}}_m^{(X,Y)} = \frac{1}{\sum_{s=1}^{S}\gamma_m^{(s)}} \sum_{s=1}^{S} \begin{bmatrix} \hat{\boldsymbol{W}}^{(s)}\langle\boldsymbol{\zeta}\rangle_m^{(s)} \\ \langle\boldsymbol{Y}\rangle_m^{(s)} \end{bmatrix} \quad (26)$$

$$\hat{\boldsymbol{\Sigma}}_m^{(X,Y)} = \frac{1}{\sum_{s=1}^{S}\gamma_m^{(s)}} \sum_{s=1}^{S} \begin{bmatrix} \hat{\boldsymbol{W}}^{(s)}\langle\boldsymbol{\zeta}\boldsymbol{\zeta}^\top\rangle_m^{(s)}\hat{\boldsymbol{W}}^{(s)^\top} & \hat{\boldsymbol{W}}^{(s)}\langle\boldsymbol{\zeta}\boldsymbol{Y}^\top\rangle_m^{(s)} \\ \langle\boldsymbol{Y}\boldsymbol{\zeta}^\top\rangle_m^{(s)}\hat{\boldsymbol{W}}^{(s)^\top} & \langle\boldsymbol{Y}\boldsymbol{Y}^\top\rangle_m^{(s)} \end{bmatrix} - \hat{\boldsymbol{\mu}}_m^{(X,Y)}\hat{\boldsymbol{\mu}}_m^{(X,Y)^\top}. \quad (27)$$

4.4. Implementation of MAP for Unsupervised Adaptation

To robustly estimate the linear transform, particularly when the amount of adaptation data is limited, the MAP estimation has successfully been applied to the CMLLR adaptation [20]. This adaptation method is called constrained MAP linear regression (CMAPLR).

The MAP estimation employs prior information of the adaptation parameters. In this paper, the prior probability density function of the extended transform is given by

$$P(\boldsymbol{W}|\lambda_W) = \prod_{i=1}^{2D} \mathcal{N}(\boldsymbol{w}_{(i)}; \boldsymbol{\mu}_i^{(W)}, \boldsymbol{\Sigma}_i^{(W)}). \quad (28)$$

The mean vector $\boldsymbol{\mu}_i^{(W)}$ and the diagonal covariance matrix $\boldsymbol{\Sigma}_i^{(W)}$ for the i^{th} row are estimated in the sense of maximum likelihood using the extended transforms for individual prestored source speakers $\hat{\boldsymbol{W}}^{(1:S)}$ determined in SAT.

In the unsupervised CMAPLR adaptation, the extended transform is determined by maximizing the posterior probability density function as

$$\hat{\boldsymbol{W}}^{(MAP)} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{X}_1^{(new)}, \cdots, \boldsymbol{X}_T^{(new)}, \boldsymbol{\lambda}, \boldsymbol{W}) = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\lambda_W)^\tau \prod_{t=1}^{T} P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \boldsymbol{W}), \quad (29)$$

where τ is a hyperparameter. The EM algorithm is used by maximizing the auxiliary function given by

$$\mathcal{Q}^{(MAP)}(\boldsymbol{W}, \hat{\boldsymbol{W}}) = \mathcal{Q}^{(ML)}(\boldsymbol{W}, \hat{\boldsymbol{W}}) + \tau \log P(\hat{\boldsymbol{W}}|\lambda_W). \quad (30)$$

The MAP estimate of the i^{th} row vector $\boldsymbol{w}_{(i)}$ of the extended transform $\boldsymbol{W}$ in the row by row optimization is given by

$$\hat{\boldsymbol{w}}_{(i)}^{(MAP)} = \left(\rho\boldsymbol{c}_i + \boldsymbol{k}_i + \tau\boldsymbol{\mu}_i^{(W)^\top}\boldsymbol{\Sigma}_i^{(W)^{-1}}\right)\left(\boldsymbol{G}_{i,i} + \tau\boldsymbol{\Sigma}_i^{(W)^{-1}}\right)^{-1}. \quad (31)$$

5. Many-to-One VC Algorithms Based on Mean Linear Transformation

In this section, linear transformation techniques based on the MLLR mean adaptation are applied to many to one VC.

5.1. MLLR Mean Adaptation for Many-to-One VC

The MLLR mean adaptation for many to one VC linearly trans forms the source mean vector as follows:

$$\tilde{\boldsymbol{\mu}}_m^{(X)} = \boldsymbol{A}'\boldsymbol{\mu}_m^{(X)} + \boldsymbol{b}' = \boldsymbol{W}'\boldsymbol{\xi}_m, \tag{32}$$

where $\boldsymbol{W}'$ is the extended MLLR transform $[\boldsymbol{b}', \boldsymbol{A}']$, and $\boldsymbol{\xi}_m$ is the extended source mean vector $\left[1, \boldsymbol{\mu}_m^{(X)\top}\right]^\top$. The joint probability density function of the adapted GMM is written as

$$P(\boldsymbol{X}_t, \boldsymbol{Y}_t|\boldsymbol{\lambda}, \boldsymbol{W}') = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\begin{bmatrix}\boldsymbol{X}_t \\ \boldsymbol{Y}_t\end{bmatrix}; \tilde{\boldsymbol{\mu}}_m^{(X,Y)}, \boldsymbol{\Sigma}_m^{(X,Y)}\right), \tag{33}$$

where the transformed mean vector of the m^{th} mixture compo nent is given by $\tilde{\boldsymbol{\mu}}_m^{(X,Y)} = \left[\tilde{\boldsymbol{\mu}}_m^{(X)\top}, \boldsymbol{\mu}_m^{(Y)\top}\right]^\top$.

5.2. Unsupervised Adaptation

The extended MLLR transform is estimated as

$$\hat{\boldsymbol{W}}' = \arg\max_{\boldsymbol{W}'} \prod_{t=1}^{T} P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \boldsymbol{W}'). \tag{34}$$

The auxiliary function maximized in the EM algorithm is given by

$$\mathcal{Q}'^{(ML)}(\boldsymbol{W}', \hat{\boldsymbol{W}}') = \sum_{t=1}^{T} \gamma_{m,t}'^{(new)} \log P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \hat{\boldsymbol{W}}') \tag{35}$$

$$\gamma_{m,t}'^{(new)} = P(m|\boldsymbol{X}_t^{(new)}, \boldsymbol{\lambda}, \boldsymbol{W}'). \tag{36}$$

In the E step, the sufficient statistics are calculated as

$$\gamma_m'^{(new)} = \sum_{t=1}^{T} \gamma_{m,t}'^{(new)} \tag{37}$$

$$\langle \boldsymbol{X} \rangle_m^{(new)} = \sum_{t=1}^{T} \gamma_{m,t}'^{(new)} \boldsymbol{X}_t^{(new)}. \tag{38}$$

In the M step, the extended MLLR transform is also updated by row by row optimization [13, 18]. The ML estimate of the i^{th} row vector $\boldsymbol{w}'_{(i)}$ of the extended MLLR transform $\boldsymbol{W}'$ is given by

$$\hat{\boldsymbol{w}}'_{(i)} = \boldsymbol{k}'_i \boldsymbol{G}'^{-1}_{i,i} \tag{39}$$

$$\boldsymbol{G}'_{i,j} = \sum_{m=1}^{M} \gamma_m'^{(new)} p_{m,(i,j)}^{(XX)} \boldsymbol{\xi}_m \boldsymbol{\xi}_m^\top \tag{40}$$

$$\boldsymbol{k}'_i = \sum_{m=1}^{M} \boldsymbol{p}_{m,(i)}^{(XX)} \langle \boldsymbol{X} \rangle_m^{(new)} \boldsymbol{\xi}_m^\top - \sum_{j=1, j\neq i}^{2D} \boldsymbol{w}'_{(j)} \boldsymbol{G}'_{i,j}. \tag{41}$$

It is worth noting that if the diagonal matrix is used as every source covariance matrix $\boldsymbol{\Sigma}_m^{(XX)}$, the ML estimate of each row vector does not depend on the other row vectors because the second term in the R.H.S. of Eq. (41) disappears. Namely, the iterative update of the extended MLLR transform is not required in the M step.

It is also possible to use multiple transforms in the MLLR mean adaptation in the same manner as that in the CMLLR adaptation.

5.3. Implementation of SAT

SAT is also available for the MLLR mean adaptation. The canonical GMM parameter set $\boldsymbol{\lambda}$ and a set of pre stored source speaker dependent linear transforms $\boldsymbol{W}'^{(1:S)} = \left\{\boldsymbol{W}'^{(1)}, \cdots, \boldsymbol{W}'^{(S)}\right\}$ are optimized as

$$\left\{\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}'^{(1:S)}\right\} = \arg\max_{\left\{\boldsymbol{\lambda}, \boldsymbol{W}'^{(1:S)}\right\}} \prod_{s=1}^{S} \prod_{t=1}^{T_s} P(\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t|\boldsymbol{\lambda}, \boldsymbol{W}'^{(s)}). \tag{42}$$

The auxiliary function maximized in the EM algorithm is given by

$$\mathcal{Q}'^{(SAT)}\left(\left\{\boldsymbol{\lambda}, \boldsymbol{W}'^{(1:S)}\right\}, \left\{\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}'^{(1:S)}\right\}\right) = \sum_{t=1}^{T} \gamma_{m,t}'^{(s)} \log P(\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t|\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{W}}'^{(s)}) \tag{43}$$

$$\gamma_{m,t}'^{(s)} = P(m|\boldsymbol{X}_t^{(s)}, \boldsymbol{Y}_t, \boldsymbol{\lambda}, \boldsymbol{W}'^{(s)}). \tag{44}$$

In the E step, the sufficient statistics are calculated as

$$\gamma_m'^{(s)} = \sum_{t=1}^{T} \gamma_{m,t}'^{(s)} \tag{45}$$

$$\langle \boldsymbol{Z} \rangle_m^{(s)} = \begin{bmatrix}\langle \boldsymbol{X} \rangle_m^{(s)} \\ \langle \boldsymbol{Y} \rangle_m^{(s)}\end{bmatrix} = \sum_{t=1}^{T} \gamma_{m,t}'^{(s)} \begin{bmatrix}\boldsymbol{X}_t^{(s)} \\ \boldsymbol{Y}_t\end{bmatrix} \tag{46}$$

$$\langle \boldsymbol{Z}\boldsymbol{Z}^\top \rangle_m^{(s)} = \sum_{t=1}^{T} \gamma_{m,t}'^{(s)} \begin{bmatrix}\boldsymbol{X}_t^{(s)} \\ \boldsymbol{Y}_t\end{bmatrix} \begin{bmatrix}\boldsymbol{X}_t^{(s)} \\ \boldsymbol{Y}_t\end{bmatrix}^\top. \tag{47}$$

In the M step, the extended transform for each prestored source speaker is updated by row by row optimization. The ML es timate of the i^{th} row vector $\boldsymbol{w}_i'^{(s)}$ of the extended transform $\boldsymbol{W}'^{(s)}$ for the s^{th} prestored source speaker is given by

$$\hat{\boldsymbol{w}}_{(i)}'^{(s)} = \boldsymbol{k}_i'^{(s)} \boldsymbol{G}_{i,i}'^{(s)^{-1}} \tag{48}$$

$$\boldsymbol{G}_{i,j}'^{(s)} = \sum_{m=1}^{M} \gamma_m'^{(s)} p_{m,(i,j)}^{(X,Y)} \boldsymbol{\xi}_m \boldsymbol{\xi}_m^\top \tag{49}$$

$$\boldsymbol{k}_i'^{(s)} = \sum_{m=1}^{M} \boldsymbol{p}_{m,(i)}^{(X,Y)} \langle \boldsymbol{Z} \rangle_m^{(s)} \boldsymbol{\xi}_m^\top - \sum_{j=1, j\neq i}^{2D} \boldsymbol{w}_{(j)}'^{(s)} \boldsymbol{G}_{i,j}'^{(s)} - \sum_{m=1}^{M} \gamma_m'^{(s)} \left(\sum_{j=1}^{2D} p_{m,(i,j+2D)}^{(X,Y)} \mu_{m,(j)}^{(Y)}\right) \boldsymbol{\xi}_m^\top. \tag{50}$$

The value $\mu_{m,(j)}^{(y)}$ is the j^{th} component of the target mean vec tor $\boldsymbol{\mu}_m^{(Y)}$. This estimation process is regarded as supervised MLLR mean adaptation with the joint feature vectors of the source and target speakers. Note that even if diagonal covari ance matrix is used as every source covariance matrix $\boldsymbol{\Sigma}_m^{(XX)}$, the iterative row by row update is still necessary because the cross covariance matrices $\boldsymbol{\Sigma}_m^{(YX)}$ affect the third term in the R.H.S. of Eq. (50). The canonical GMM parameter set is up dated as

$$\hat{\alpha}_m = \frac{1}{\sum_{m=1}^{M}\sum_{s=1}^{S} \gamma_m'^{(s)}} \sum_{s=1}^{S} \gamma_m'^{(s)} \tag{51}$$

$$\hat{\boldsymbol{\mu}}_m^{(X,Y)} = \left(\sum_{s=1}^{S} \gamma_m'^{(s)} \begin{bmatrix}\hat{\boldsymbol{A}}'^{(s)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^\top \boldsymbol{\Sigma}_m^{(X,Y)^{-1}} \begin{bmatrix}\hat{\boldsymbol{A}}'^{(s)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\right)^{-1} \cdot \sum_{s=1}^{S} \begin{bmatrix}\hat{\boldsymbol{A}}'^{(s)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^\top \boldsymbol{\Sigma}_m^{(X,Y)^{-1}} \begin{bmatrix}\langle \boldsymbol{X} \rangle_m^{(s)} - \gamma_m'^{(s)} \boldsymbol{b}'^{(s)} \\ \langle \boldsymbol{Y} \rangle_m^{(s)}\end{bmatrix} \tag{52}$$

$$\hat{\boldsymbol{\Sigma}}_m^{(X,Y)} = \frac{1}{\sum_{s=1}^{S}\gamma_m'^{(s)}}\sum_{s=1}^{S}\Bigg\{\langle \boldsymbol{Z}\boldsymbol{Z}^\top\rangle_m^{(s)} - \langle \boldsymbol{Z}\rangle_m^{(s)}\begin{bmatrix}\hat{\boldsymbol{W}}'^{(s)}\hat{\boldsymbol{\xi}}_m \\ \hat{\boldsymbol{\mu}}_m^{(Y)}\end{bmatrix}^\top - \begin{bmatrix}\hat{\boldsymbol{W}}'^{(s)}\hat{\boldsymbol{\xi}}_m \\ \hat{\boldsymbol{\mu}}_m^{(Y)}\end{bmatrix}\langle \boldsymbol{Z}\rangle_m^{(s)\top} + \gamma_m'^{(s)}\begin{bmatrix}\hat{\boldsymbol{W}}'^{(s)}\hat{\boldsymbol{\xi}}_m \\ \hat{\boldsymbol{\mu}}_m^{(Y)}\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{W}}'^{(s)}\hat{\boldsymbol{\xi}}_m \\ \hat{\boldsymbol{\mu}}_m^{(Y)}\end{bmatrix}^\top\Bigg\}, \quad (53)$$

where $\hat{\boldsymbol{\xi}}_m$ is the extended mean vector given by $\left[1, \hat{\boldsymbol{\mu}}_m^{(X)\top}\right]^\top$. Note that mean vectors and covariance matrices are updated sequentially because their ML estimates depend on each other.

5.4. Implementation of MAP for Unsupervised Adaptation

The MAP estimation has also successfully been applied to the MLLR mean adaptation [21]. This adaptation method is called MAP linear regression (MAPLR).

The unsupervised MAPLR adaptation estimates the extended transform as

$$\hat{\boldsymbol{W}}'^{(MAP)} = \arg\max_{\boldsymbol{W}'} P(\boldsymbol{W}'|\lambda_W)^\tau \prod_{t=1}^{T} P(\boldsymbol{X}_t^{(new)}|\boldsymbol{\lambda}, \boldsymbol{W}'), \quad (54)$$

where the prior probability density function is defined in the same manner as that in the CMAPLR adaptation. The EM algorithm is used by maximizing the auxiliary function given by

$$\mathcal{Q}'^{(MAP)}(\boldsymbol{W}', \hat{\boldsymbol{W}}') = \mathcal{Q}'^{(ML)}(\boldsymbol{W}', \hat{\boldsymbol{W}}') + \tau \log P(\hat{\boldsymbol{W}}'|\lambda_W). \quad (55)$$

The MAP estimate of the i^{th} row vector $\boldsymbol{w}_i'$ of $\boldsymbol{W}'$ is given by

$$\hat{\boldsymbol{w}}_{(i)}'^{(MAP)} = \left(\boldsymbol{k}_i' + \tau \boldsymbol{\mu}_i^{(W)\top}\boldsymbol{\Sigma}_i^{(W)^{-1}}\right)\left(\boldsymbol{G}_{i,i}' + \tau\boldsymbol{\Sigma}_i^{(W)^{-1}}\right)^{-1}. \quad (56)$$

6. Implementation Issues in VC

6.1. Computational Cost

In unsupervised adaptation, the MLLR mean adaptation is much more computationally efficient than the CMLLR adaptation. In the E step, the CMLLR adaptation should calculate the second order sufficient statistics in Eq. (12); however, these statistics are unnecessary in the MLLR mean adaptation. Moreover, the MLLR mean adaptation requires a much lower computational cost to update each row vector in the M step than the CMLLR adaptation because the cofactor calculation is not necessary; furthermore, even iterative updates in the row by row optimization are not necessary if the source covariance matrices are diagonal, as mentioned above.

In conversion, the computational cost does not change before and after the MLLR mean adaptation. On the other hand, it changes in the CMLLR adaptation if the block covariance matrices of the GMM are originally diagonal. Since the transform is not diagonal in general, the adapted block covariance matrices in Eq. (6) are full. The use of such matrices results in a much higher computational cost. Alternatively, an efficient way is to apply the CMLLR transform to the source feature vector rather than the model parameters as

$$P(\boldsymbol{X}_t, \boldsymbol{Y}_t|\boldsymbol{\lambda}, \boldsymbol{W}) = \sqrt{|\boldsymbol{A}|^2}P(\tilde{\boldsymbol{X}}_t, \boldsymbol{Y}_t|\boldsymbol{\lambda}). \quad (57)$$

The transformed source feature vectors are converted with the original GMM parameters. Thus, only the additional computational cost is necessary to transform the source feature vector.

6.2. Numerical Accuracy

One implementation issue has to be resolved, particularly in the MLLR mean adaptation with multiple transforms. If the number of mixture components in eash regression class is less than the dimensionality of $\boldsymbol{w}_{(i)}'$, the matrix $\boldsymbol{G}_{i,i}'$ in Eq. (39) cannot have a full rank. This situation easily arises in the GMM based VC. The use of a block diagonal matrix or a band matrix as the transform can effectively reduce the dimensionality of $\boldsymbol{w}_{(i)}'$. Moreover, MAPLR effectively resolves this issue by adding the precision matrix $\boldsymbol{\Sigma}_i^{(W)^{-1}}$ of the prior probability density function to $\boldsymbol{G}_{i,i}'$, as shown in Eq. (56). On the other hand, in the CMLLR adaptation, this rank deficiency problem rarely occurs because the rank of $\boldsymbol{G}_{i,i}$ in Eq. (13) depends on the number of frames in adaptation data.

7. Experimental Evaluations

7.1. Experimental Conditions

As the prestored source speakers, 80 male and 80 female speakers were used. Each of these speakers uttered a set of 50 phonetically balanced sentences from the seven different sets. The target speaker was another male speaker who uttered all of these sets. As new source speakers to be adapted, five male and five female speakers not included in the prestored source speakers were used. Each of these speakers uttered 53 sentences also not included in the prestored data sets. The number of adaptation sentences was varied from 1 to 32. The remaining 21 sentences were used as the evaluation data. All speech data were sampled at 16 kHz. The 1$^{\text{st}}$ through 24$^{\text{th}}$ mel cepstral coefficients were used as spectral features. More detailed conditions are described in [7, 10].

The proposed linear transformation approaches were compared with each other. Mel cepstral distortion between the converted and target mel cepstra was used as an evaluation metric. To evaluate only the effect of the GMM on the spectral conversion, the GV was not considered in the conversion process. The number of mixture components was set to 128. The block covariance matrices $\boldsymbol{\Sigma}_m^{(XX)}$, $\boldsymbol{\Sigma}_m^{(XY)}$, $\boldsymbol{\Sigma}_m^{(YX)}$, and $\boldsymbol{\Sigma}_m^{(YY)}$ were set to diagonal matrices. A block diagonal matrix consisting of two square matrices for static and dynamic parts was used as the linear transformation matrix. The hyperparameter τ in CMAPLR and MAPLR was manually determined.

7.2. Experimental Results

Figure 1 shows the results of unsupervised adaptation using a single transform as well as the SI GMM without adaptation. Mel cepstral distortion decreases with an increase in adaptation data size up to 10 adaptation utterances in both the CMLLR adaptation and the MLLR mean adaptation. However, even if 32 adaptation utterances are used, the conversion performance of the adapted models is still close to that of the SI GMM. SAT effectively decreases mel cepstral distortion by approximately 0.1 dB over any amount of adaptation data. Moreover, MAP adaptation further decreases it by approximately 0.1 dB when the amount of adaptation data is small. Consequently, the proposed methods with both SAT and MAP adaptation obviously outperform the SI GMM. The performance of the MLLR mean adaptation is comparable to that of the CMLLR adaptation, although it does not adapt the covariance matrices.

Figure 2 shows the effects of multiple linear transforms in not only the unsupervised adaptation but also the supervised adaptation with parallel adaptation data. The result of the traditional VC with 32 parallel sentences is also shown as "SD

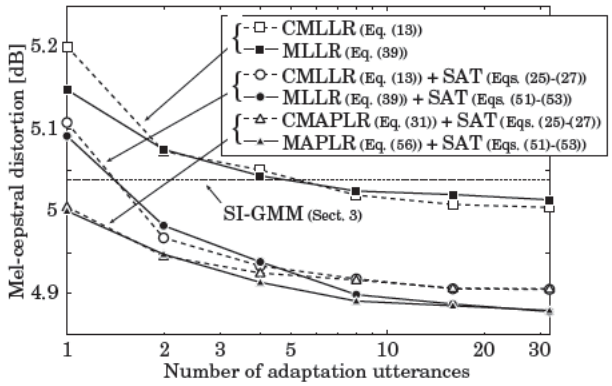

Figure 1: *Mel cepstral distortion as a function of the number of adaptation utterances in unsupervised adaptation with a single transform.*

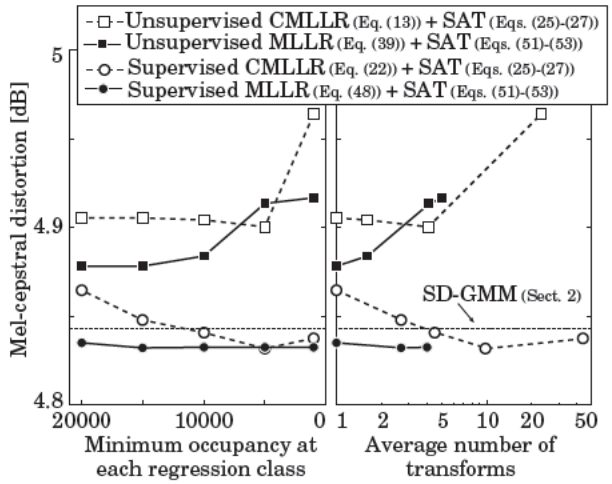

Figure 2: *Mel cepstral distortion as a function of minimum oc cupancy count (left) and as a function of the number of trans forms (right) when 32 adaptation utterances are used.*

GMM." In the MLLR mean adaptation, the minimum number of mixture components as well as the minimum occupancy count is used to dynamically determine the number of transforms to eliminate the rank deficiency problem. The use of multiple lin ear transforms causes the performance degradation in the un supervised adaptation, although it works reasonably well in the supervised adaptation. The performance differences between the supervised and unsupervised adaptations show that the esti mation of the linear transformation using only the source feature vectors is difficult even if a single transform is used. Multiple transforms are more difficult to estimate owing to their higher complexity, and therefore, their effectiveness is not observable in the unsupervised adaptation. In the supervised MLLR mean adaptation, a single transform yields the same conversion accu racy as multiple transforms. In addition, it yields higher conver sion accuracy than the SD GMM. Therefore, the use of multiple transforms is unnecessary.

8. Conclusions

In this paper, we applied linear transformation techniques to many to one voice conversion (VC). The obtained experimental results suggested that the mean adaptation based on the maxi mum *a posteriori* linear regression with a single transform and speaker adaptive training is the most effective method for un supervised adaptation in many to one VC in views of both ac curacy and efficiency among the proposed linear transformation algorithms.

Acknowledgements: This research was supported in part by MIC SCOPE and MEXT Grant in Aid for Young Scientists (A).

9. References

[1] Y. Stylianou, O. Cappé, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. SAP*, Vol. 6, No. 2, pp. 131 142, 1998.

[2] A. Mouchtaris, J.V. der Spiegel, and P. Mueller. Nonparallel training for voice conversion based on a parameter adaptation approach. *IEEE Trans. ASLP*, Vol. 14, No. 3, pp. 952 963, 2006.

[3] V.D. Diakoloukas and V.V. Digalakis. Maximum-likelihood stochastic-transformation adaptation of hidden Markov models. *IEEE Trans. SAP*, Vol. 7, No. 2, pp. 177 187, 1999.

[4] C.-H. Lee and C.-H. Wu. MAP-based adaptation for speech conversion using adaptation data selection and non-parallel training. *Proc. INTERSPEECH*, pp. 2446 2449, Pittsburgh, USA, Sep. 2006.

[5] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. SAP*, Vol. 2, No. 2, pp. 291 298, 1994.

[6] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu. Voice conversion with smoothed GMM and MAP adaptation. *Proc. INTERSPEECH*, pp. 2413 2416, Geneva, Switzerland, Sep. 2003.

[7] T. Toda, Y. Ohtani, and K. Shikano. One-to-many and many-to-one voice conversion based on eigenvoices. *Proc. ICASSP*, pp. 1249 1252, Hawaii, USA, Apr. 2007.

[8] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. SAP*, Vol. 8, No. 6, pp. 695 707, 2000.

[9] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano. Many-to-many eigenvoice conversion with reference voice. *Proc. INTERSPEECH*, pp. 1623 1626, Brighton, UK, Sep. 2009.

[10] D. Tani, T. Toda, Y. Ohtani, H. Saruwatari, and K. Shikano. Maximum a posteriori adaptation for many-to-one eigenvoice conversion. *Proc. INTERSPEECH*, pp. 1461 1464, Brisbane, Australia, Sep. 2008.

[11] 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, 1995.

[12] V.V. Digalakis, D. Rtischev, and L.G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Trans. SAP*, Vol. 3, No. 5, pp. 357 366, 1995.

[13] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, Vol. 12, No. 2, pp. 75 98, 1998.

[14] D. Miyamoto, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano. Acoustic compensation methods for body transmitted speech conversion. *Proc. ICASSP*, pp. 3901 3904, Taipei, Taiwan, Apr. 2009.

[15] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. *Proc. ICSLP*, pp. 1137 1140, Philadelphia, USA, Oct. 1996.

[16] A. Kain and M.W. Macon. Spectral voice conversion for text-to-speech synthesis. *Proc. ICASSP*, pp. 285 288, Seattle, USA, May 1998.

[17] T. Toda, A.W. Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Trans. ASLP*, Vol. 15, No. 8, pp. 2222 2235, 2007.

[18] K.C. Sim and M.J.F. Gales. Adaptation of precision matrix models on large vocabulary continuous speech recognition. *Proc. ICASSP*, Vol. 1, pp. 97 100, Philadelphia, USA, Mar. 2005.

[19] M.J.F. Gales. The generation and use of regression class trees for MLLR adaptation. *Technical Report*, CUED/F-INFENG/TR263, Cambridge University, 1996.

[20] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai. Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. *IEEE Trans. ASLP*, Vol. 17, No. 1, pp. 66 83, 2009.

[21] O. Siohan, T.A. Myrvoll, and C.-H. Lee. Structual maximum a posteriori linear regression for fast HMM adaptation. *Computer Speech and Language*, Vol. 16, No. 1, pp. 5 24, 2002.