Voice Conversion Using Partial Least Squares Regression

Elina Helander, Tuomas Virtanen, *Member, IEEE,* Jani Nurminen, and Moncef Gabbouj, *Senior Member, IEEE*

***Abstract*—Voice conversion can be formulated as finding a mapping function which transforms the features of the source speaker to those of the target speaker. Gaussian mixture model (GMM) based conversion is commonly used, but it is subject to overfitting. In this paper, we propose to use partial least squares (PLS) based transforms in voice conversion. To prevent overfitting, the degrees of freedom in the mapping can be controlled by choosing a suitable number of components. We propose a technique to combine PLS with GMMs, enabling the use of multiple local linear mappings. To further improve the perceptual quality of the mapping where rapid transitions between GMM components produce audible artefacts, we propose to low-pass filter the component posterior probabilities.**

The conducted experiments show that the proposed technique results in better subjective and objective quality than the baseline joint density GMM approach. In speech quality conversion preference tests, the proposed method achieved 67% preference score against the smoothed joint density GMM method and 84% preference score against the unsmoothed joint density GMM method. In objective tests the proposed method produced a lower Mel-cepstral distortion than the reference methods.

I. Introduction

FEATURE transformation refers to a process where features from one domain are mapped to another domain in a desired way. In the area of speech processing, feature transformation techniques can be utilized in many applications, such as bandwidth extension of narrowband speech [1], emotional conversion [2], and single-channel enhancement [3], but perhaps the most evident application is *voice conversion* (VC). The goal in voice conversion is to modify speech spoken by one speaker (*source*) to give an impression that it was spoken by another specific speaker (*target*). The features to be transformed in voice conversion can be any parameters describing the speech and the speaker, including segmental cues in the spectral envelope and suprasegmental cues such as the fundamental frequency F_0 and phoneme durations.

Among the different applications of feature transformation, voice conversion involves some unique properties that make it a challenging process. First, the conversion result should fulfill the sometimes contradictory goals related to speech quality and the success of identity conversion. Second, the amount of data available for training is often rather limited in practical use cases. Additional challenges are also caused by the fact that the perception of quality is largely subjective, although objective quality measures approximating the subjective rating have been proposed [4]. To add another degree of complexity to the problem, the same person can utter the same sentence in multiple different ways. Due to the above-mentioned fact and the lack of high-quality objective measures, listening tests should always be used in the development and evaluation of voice conversion systems.

A conventional problem formulation in voice conversion involves a source speaker A whose speech characteristics are to be transformed to resemble the speaker characteristics of the target speaker B as closely as possible. A prerequisite for building any type of voice conversion model is that there has to be a certain amount of training data available from both the source speaker and the target speaker. The requirements for the training data sizes are system specific, and the data can be either parallel, i.e. the speakers have uttered the same sentences, or non-parallel. The sentences spoken by the speakers can be known or unknown, corresponding to the cases of text dependent and text independent voice conversion. The most extreme case of text independent voice conversion is cross-lingual conversion [5] where A and B speak different languages that may even have different phoneme sets.

The performance of a voice conversion system is typically rather heavily dependent on the speaker pair A-B, which creates large variations in the observed quality. This issue has been tackled at least partly in average-speaker hidden Markov model (HMM)-based speech synthesis [6], [7] or through the use of eigenvoices [8]. In HMM-based speech synthesis, voice adaptation enables mimicking of new voices using only small training data sets in a manner similar to that of speech recognition. The average voice model has been found to effectively serve as a "source speaker" [6]. In the eigenvoice approach, also originally developed for speaker adaptation [9], it is assumed that the parameters of any speaker can be formed as a linear combination of eigenvoices. The eigenvoices can capture speaker variations effectively. There are, however, factors that limit the general usefulness of the above approaches, i.e. the average voice model requires an HMM-based speech synthesis system, and in the eigenvoice approach, a large number of pre-stored speakers with parallel training data must be available. In general, the requirement of having large amounts of parallel training data is prohibitive and there have also been proposals for coping with non-parallel data [10].

The parameterization of the speech data and the flexibility of the analysis/synthesis framework play an important role

E. Helander, T. Virtanen and M. Gabbouj are with the Department of Signal Processing, Tampere University of Technology, Korkeakoulunkatu 1, Tampere, Finland, e-mail: (elina.helander@tut.fi; tuomas.virtanen@tut.fi, moncef.gabbouj@tut.fi), phone: +358 3 3115 4798, fax: +358 3 3115 3966. J. Nurminen is with Nokia Devices R&D, Tampere, Finland, e-mail: jani.k.nurminen@nokia.com. This work was supported by the Academy of Finland (application number 129657, Finnish Programme for Centres of Excellence in Research 2006-2011). In addition, E. Helander was supported by the graduate school of Tampere University of Technology.

in the final speech quality of converted speech. There is no straightforward solution to the parameterization problem. For example, speaker identity could be conveniently characterized in part with formant heights and positions but robust estimation of formants is, however, difficult. The most common features used for modeling spectral content in voice conversion are based on the direct use of spectral bands or on the source-filter theory. Examples of typical features include MFCCs (Mel-frequency cepstral coefficients) (e.g. [11]), LSFs (line spectral frequencies) (e.g. [12], [13]) or MGCs (mel-generalized cepstral coefficients) (e.g. [14]).

In addition to the information on speaker identity that can be represented using spectral features, speech prosody includes important cues of identity. Compared to the attention attracted by short-term spectral conversion, there has not been much research done on converting prosodic features such as F_0 movements and speaking rhythm. Prosody is a supra-segmental phenomenon that is not conveyed through a single phonetic segment but through larger units as words, sentences, utterances or even paragraphs. Perhaps due to this reason, prosodic modeling in identity conversion has often been neglected. In most cases, only simple statistical mean and variance based F_0 conversion methods are applied, sometimes together with average speaking rate modification. More detailed prosody conversion techniques have been proposed for example in [15], [16], [17].

Mapping the source spectral envelope into the target spectral envelope has gained a lot of interest. The most common approaches are based on codebook mapping (e.g. [13], [18]) or acoustic space modeling with Gaussian mixture models ([11], [12]). The former approach is prone to errors due to discontinuity, and in addition, the amount of training data must be rather high in order to guarantee good quality. The latter approach, which uses GMMs, has been found to offer a reasonably good performance. On the other hand, the well-known drawbacks of GMM based conversion are oversmoothing and overfitting. In addition, another problem is that GMM-based conversion is time-independent. An approach for solving the time-independency and over-smoothing problems was proposed in [14] through the introduction of maximum likelihood estimation (MLE) of spectral parameter trajectory and retention of the global variance of the original parameters similarly as in HMM-based speech synthesis. To overcome the speech quality problems, the usage of frequency warping with a GMM was proposed in [19]. The warping itself does not introduce much distortion but the quality of identity conversion is typically poor, leaving the method insufficient for many potential applications. In an attempt to overcome this limitation, combining the frequency-warped source spectra with parts of the target spectra selected from the training data has been proposed in [20].

A fundamental problem in VC is how to find a proper balance between simple and complex models, especially when the amount of training data is limited. This problem is common for all regression and model fitting tasks, and it is also referred to as bias-variance dilemma [21]. In essence, simple VC models may not be able to capture the underlying relationships between the source and the target data and are typically subject to oversmoothing, whereas the use of complex models may easily result in overfitting. Overfitting occurs when a model has too many degrees of freedom compared to the amount of training data available. Overfitting results in poor prediction ability on new data while giving very good results for the training data; small fluctuations in the data become over-emphasized.

In GMM-based VC overfitting can be caused by two factors: first, the GMMs may be overfitted to the training set. Second, when a separate mapping function is estimated, it may also become overfitted. In particular, GMMs with full covariance matrices are difficult to estimate and are subject to overfitting as illustrated in [22]. Using full covariance matrices in GMM-based conversion poses the requirement of large training data sizes but an effective representation can be formed with a reasonably low number of mixtures. In contrast, a high number of mixtures is required for accurate parameter modeling with simple diagonal covariance matrices. Considering these problems, a mixture of factor analyzers was applied in [23]. Alternatively, a source GMM can be built from a larger data set and only the means are adapted using maximum a posteriori estimation [24]. Also for speaker identification, it is common to adapt only the means [25].

In this paper, we propose to use partial least squares (PLS) to obtain a mapping function between the source and the target. PLS is a regression method which specifically addresses the cross-correlation between the predictor and predicted variables. It also addresses possible collinearity of the data which is important in applications with many variables and few observations such as chemometrics, functional brain imaging and genomic analysis. For the underlying voice conversion application, the most important property of PLS is its good performance on new data with only a small amount of training observations. As we acknowledge that a single linear transform is not effective for all the source data, we formulate the use of PLS with a source GMM in a manner similar to [11], and demonstrate the effectiveness of this approach. The use of PLS can be thought as an intermediate approach between the diagonal and full covariance matrix GMM conversion. Alternatively, it can be used instead of a standard multivariate regression in a codebook based mapping, similar as in [26], or for example with fuzzy k-means.

To prevent problems with full-covariance GMMs, we use diagonal covariances for a source GMM and derive a mapping function to model the relationship between the source and the target. An ideal mapping function would be powerful enough to represent the underlying relationships of the data, but not so powerful that it slavishly models the noise associated with the data. We assume that the underlying relationship between the source and the target features can be explained by fewer variables than with full matrix transforms. Full transforms can end up modeling relationships that are actually noise.

This paper is organized as follows. Section II describes the conventional GMM-based conversion with either full or diagonal covariance matrices. In Section III, we describe the partial least squares method and extend it to the GMM-based voice conversion. Section IV describes a method for smoothing GMM posterior probabilities for improving the subjective quality of the converted speech. Practical experiments and the results are described in Section V. Some discussion is provided in Section VI. Finally, Section VII concludes the paper.

II. GMM-BASED FEATURE MODELING AND TRANSFORMATION

Voice conversion can be defined as mapping the source feature vector $\mathbf{x}_t$ into the target feature vector $\mathbf{y}_t$, at each time t. The conversion function $\hat{\mathbf{y}}_t = \mathcal{F}(\mathbf{x}_t)$ is found by minimizing the sum of squared conversion errors over all T pairs of training samples $(\mathbf{x}_t, \mathbf{y}_t)$, $t = 1, \ldots, T$ [11], given as

$$e = \sum_{t=1}^{T} \|\mathbf{y}_t - \mathcal{F}(\mathbf{x}_t)\|^2. \tag{1}$$

Linear conversion functions have been found to produce good results, but using a single global transformation limits the performance significantly. A practical solution is to model the data using a Gaussian mixture model and to find a local linear transformation for each Gaussian. Two approaches are mainly used: modeling the source with a GMM [11] or modeling the joint density between speakers [12], the latter being more popular. In the joint density model, the conversion function can be obtained directly from the GMM parameters and in the source density model, a mapping function using least-squares estimation must be estimated. The conversion function is actually a weighted sum of linear regression models. We will briefly review both approaches in the context of full and diagonal covariance matrices.

A. Source GMM model

The distribution of the source spectral vectors is modeled with a GMM as follows

$$p(\mathbf{x}_t) = \sum_{n=1}^{M} \alpha_n \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \tag{2}$$

where α_n is the prior probability of Gaussian $n = 1 \ldots M$ and $\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ is the multivariate normal distribution with mean vector $\boldsymbol{\mu}_n$, and covariance matrix $\boldsymbol{\Sigma}_n$.

The conversion function is typically assumed to be linear for each Gaussian, i.e., it has the form

$$\mathcal{F}(\mathbf{x}_t) = \sum_{n=1}^{M} \omega_{n,t}(\boldsymbol{\beta}_n \mathbf{x}_t + \mathbf{b}_n), \tag{3}$$

where $\boldsymbol{\beta}_n$ is the linear transform matrix for samples in cluster n and $\mathbf{b}_n$ is a static bias vector. Observation-dependent weights $\omega_{n,t}$ are the posterior probabilities that the nth Gaussian has produced observation $\mathbf{x}_t$ and expressed as

$$\omega_{n,t} = \frac{\alpha_n \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)}{\sum_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}. \tag{4}$$

A least-squares solution for the linear mapping of the form (3) was proposed by Stylianou et al. in [11]. Given training data pairs $(\mathbf{x}_t, \mathbf{y}_t)$, $t = 1, \ldots, T$, the solution for all $\boldsymbol{\beta}_n$ and $\mathbf{b}_n$ is found by solving a set of normal equations. In the case of diagonal GMM, the conversion function was defined in each feature dimension separately in the original work of Stylianou et al. [11]. However, in our work the conversion function of a diagonal GMM is trained jointly for all the dimensions to model the dependencies between all the source and the target features.

B. Joint density model

In the joint density model [12], the source vectors $\mathbf{x}_t$ are augmented with the corresponding target features $\mathbf{y}_t$ as $\mathbf{z}_t = [\mathbf{x}_t^T \mathbf{y}_t^T]^T$ and the GMM is estimated for the augmented vectors. The means and covariances of the GMM of the augmented vectors are given as

$$\boldsymbol{\mu}_n^z = \begin{bmatrix} \boldsymbol{\mu}_n^x \\ \boldsymbol{\mu}_n^y \end{bmatrix} \tag{5}$$

and

$$\boldsymbol{\Sigma}_n^z = \begin{bmatrix} \boldsymbol{\Sigma}_n^{xx} & \boldsymbol{\Sigma}_n^{xy} \\ \boldsymbol{\Sigma}_n^{yx} & \boldsymbol{\Sigma}_n^{yy} \end{bmatrix}, \tag{6}$$

where vectors $\boldsymbol{\mu}_n^x$ and $\boldsymbol{\mu}_n^y$ denote the mean of the source and target entries of the augmented vector in Gaussian n, respectively, and the superscripts of the covariance matrices denote their respective covariances and cross-covariances.

In the conversion, the mapped target $\hat{\mathbf{y}}_t$ is formed from the source vector $\mathbf{x}_t$ as

$$\hat{\mathbf{y}}_t = \sum_{n=1}^{M} \omega_{n,t}[\boldsymbol{\mu}_n^y + \boldsymbol{\Sigma}_n^{yx}(\boldsymbol{\Sigma}_n^{xx})^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_n^x)] \tag{7}$$

where $\omega_{n,t}$ is the posterior probability that the nth Gaussian has produced the tth observation, calculated similarly as (4) using the source vector $\mathbf{x}_t$ and means $\boldsymbol{\mu}_n^x$ and covariances $\boldsymbol{\Sigma}_n^{xx}$.

The joint density mapping (7) is the maximum likelihood estimate of the target vectors given the source vectors. In practise the terms $\boldsymbol{\Sigma}_n^{yx}(\boldsymbol{\Sigma}_n^{xx})^{-1}$ can become very small, resulting in oversmoothed speech as reported in [24]. Furthermore, when features within a cluster are linearly dependent, the covariance matrix becomes singular so that the inverse does not exist and the method cannot be used.

In general, estimating full covariance matrices in a mixture models is a difficult problem, especially when the amount of training data is small. Diagonal covariance matrices represent a commonly used simplified alternative [8]. They lead to transforming each entry of the source vector independently from the others, which limits the conversion quality. Another solution has been proposed in [23], where the covariance structure in (6) was modeled using mixtures of factor analyzers. Factor analysis is based on the assumption that the data has been generated by a set of latent variables, and is therefore slightly similar to the method we propose in Section III. However, in [23] the transforms are determined by the distributions, whereas in our approach the transforms are independent from the distribution of the parameters. Furthermore, we take into account the interaction between each local transform by estimating them jointly. Factor analysis generally searches the informative directions in the factor space of the predictor variables (source features) but may not be highly predictive on the responses (target features).

III. Partial Least Squares

To overcome the assumption of variable independence in the diagonal-covariance joint density model or the overfitting problem in the full least squares solution, we propose to use partial least squares (PLS) regression [27] in the transformation. PLS is a technique that combines principles from principal component analysis and multivariate regression (MVR), and it is most useful in cases where the feature dimensionality of $\mathbf{x}_t$ is high and the features exhibit multicollinearity. As MVR directly operates on the relationships in the data, the underlying assumption of PLS methods is that the observed variable $\mathbf{x}_t$ is generated by a small number of latent variables which explain most of the variation in the target $\mathbf{y}_t$.

We first formulate the global model as linear transforms for source and target speaker from speaker-independent latent parameters, and then extend the model for GMMs and multiple local transforms.

Global PLS is based on the assumption that the source vector $\mathbf{x}_t$ and the target vector $\mathbf{y}_t$ are produced by a linear transformation of a speaker-independent latent variable vector $\mathbf{r}_t$ as

$$\mathbf{x}_t = \mathbf{Q}\mathbf{r}_t + \mathbf{e}_t^x \tag{8}$$

$$\mathbf{y}_t = \mathbf{P}\mathbf{r}_t + \mathbf{e}_t^y \tag{9}$$

where $\mathbf{Q}$ and $\mathbf{P}$ are speaker-specific transform matrices, and $\mathbf{e}_t^x$ and $\mathbf{e}_t^y$ are residual terms which cannot be modeled by the linear model.

Solving $\mathbf{Q}$ and $\mathbf{P}$ (see Section III-A for description of an algorithm) leads to the regression model

$$\mathbf{y}_t = \boldsymbol{\beta}\mathbf{x}_t + \mathbf{e}_t \tag{10}$$

where $\boldsymbol{\beta}$ is the regression matrix which depends on $\mathbf{Q}$ and $\mathbf{P}$, and $\mathbf{e}_t$ is the regression residual. PLS differs from the standard multivariate regression in the sense that also $\mathbf{x}_t$ is assumed to have a stochastic residual term. Furthermore, the rank of the regression matrix $\boldsymbol{\beta}$ is the dimensionality of the latent variable vector $\mathbf{r}_t$. This dimension is called the number of PLS components, and selecting it appropriately prevents overfitting effectively. The PLS model becomes equivalent to the multivariate regression if $\boldsymbol{\beta}$ has full rank, i.e., the number of latent variables equals the number of source variables.

Figure 1 illustrates the cumulative variance of the source variables (8) and the corresponding target variables (9) with different numbers of PLS components without the residual terms $\mathbf{e}_t^x$ and $\mathbf{e}_t^y$. Both the source and target vectors are 24-dimensional MGCs and the amount of data is 200 frames. It can be seen that increasing the number of PLS components increases the explained variance of predicted variables, and the source variables become perfectly explained by the PLS model when the number of PLS components equals the dimensionality of the source vector. Not all the variations in the target vectors are explained by the model even when all the PLS components are used because the target data cannot be perfectly explained as a linear combination of source variables.

Figure 2 illustrates the mean squared prediction error of a 4-fold cross-validation experiment where three fourths of the above data was used to estimate the transform and the rest were used to measure the error. Increasing the number of PLS components up to nine reduces the error, after which the error increases slightly because of overfitting. In section III-B we propose a technique where multiple local transforms are estimated together with PLS to avoid overfitting.

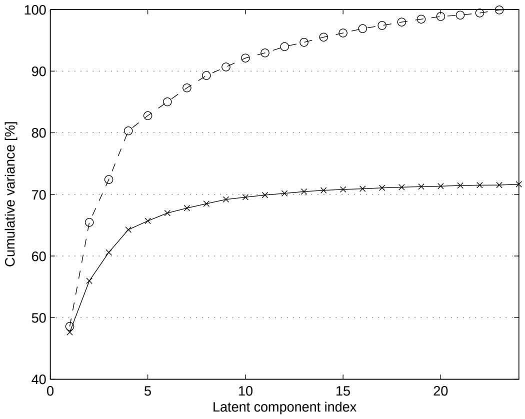

Fig. 1. *Cumulative relative variance of source variables (dashed line with circles) and target variables (solid line with x-marks) explained by the PLS model as a function of the number of PLS components.*

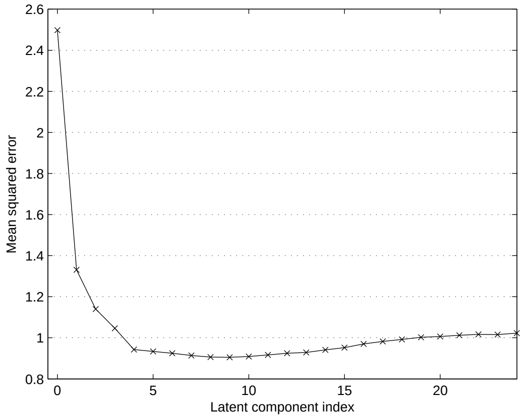

Fig. 2. *Mean squared error of the example test data as the function of PLS components. See the text for a detailed explanation.*

A. Algorithm for PLS

There exists many variants for solving the PLS regression problem. In this paper, we use the SIMPLS (simple partial least squares) algorithm proposed by de Jong [27], which has the advantages of being computationally efficient, its avoidance of matrix inverses, and operation on the original data instead of its covariances. Below is a brief description of the processing steps of the algorithm.

The algorithm operates on zero-mean source and target vectors $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively, so the empirical means of the vectors are subtracted prior to the processing, and afterwards added to the regression results. Let us denote the set of source observations by matrix $\mathbf{X} = \big[\mathbf{x}_1, \ldots, \mathbf{x}_T\big]$, and the set of target vectors by matrix $\mathbf{Y} = \big[\mathbf{y}_1, \ldots, \mathbf{y}_T\big]$. In each iteration i, the algorithm estimates score vector $\mathbf{t}$ which explains most of the cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$. The tth entry in vector $\mathbf{t}$ corresponds to a coefficient in the latent variable vector $\mathbf{r}_t$ in Eqs. (8) and (9), whereas the loading vectors $\mathbf{p}$ and $\mathbf{q}$ are the corresponding rows in matrices $\mathbf{Q}$ and $\mathbf{P}$. After each iteration, the contribution of the estimated PLS component is subtracted from the cross-covariance matrix $\mathbf{C}$. For details of the algorithm, see [27].

```
1) Initialize $\mathbf{R}$,$\mathbf{V}$, $\mathbf{Q}$, and $\mathbf{T}$ to empty matrices.
2) Calculate the cross-covariance matrix between $\mathbf{x}$ and $\mathbf{y}$ as $\mathbf{C} = \mathbf{X}\mathbf{Y}^T$.
3) Calculate the eigenvector $\mathbf{q}$ corresponding to the largest eigenvalue of $\mathbf{C}^T\mathbf{C}$.
4) Set $\mathbf{r} = \mathbf{C}\mathbf{q}$ and $\mathbf{t} = \mathbf{X}^T\mathbf{r}$.
5) Subtract the mean of its entries from $\mathbf{t}$.
6) Normalize $\mathbf{r}$ and $\mathbf{t}$ by $\mathbf{r} = \mathbf{r}/||\mathbf{t}||$ and $\mathbf{t} = \mathbf{t}/||\mathbf{t}||$.
7) Set $\mathbf{p} = \mathbf{X}\mathbf{t}$, $\mathbf{q} = \mathbf{Y}\mathbf{t}$, and $\mathbf{u} = \mathbf{Y}^T\mathbf{q}$.
8) Set $\mathbf{v} = \mathbf{p}$.
9) If iteration count $i > 1$ then orthogonalize the terms by $\mathbf{v} = \mathbf{v} - \mathbf{V}\mathbf{V}^T\mathbf{p}$ and $\mathbf{u} = \mathbf{u} - \mathbf{T}\mathbf{T}^T\mathbf{u}$.
10) Normalize $\mathbf{v}$ as $\mathbf{v} = \mathbf{v}/||\mathbf{v}||$.
11) Set $\mathbf{C} = \mathbf{C} - \mathbf{v}\mathbf{v}^T\mathbf{C}$.
12) Assign $\mathbf{r}$, $\mathbf{q}$, $\mathbf{v}$ and $\mathbf{t}$ as the $i^{\text{th}}$ columns of matrices $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{V}$ and $\mathbf{T}$, respectively.
```

The processing steps 2-12 are repeated for iterations $i = 1, 2, \ldots$ up to the number of PLS components. The number of components can be selected by crossvalidation, bootstrapping or manually by the user. The regression matrix $\boldsymbol{\beta}$ is obtained as $\boldsymbol{\beta} = \mathbf{R}\mathbf{Q}^T$.

B. Combining PLS with GMM

Similarly to the methods described in Section II, it is unlikely that a single linear transform is effective for all the data. To overcome this limitation we extend the PLS model for GMMs and multiple local transforms. A locally weighted PLS (LWPLS) algorithm was proposed in [28], where it was observed that globally high-dimensional data can be modeled using a low number of latent variables, and LWPLS can approximate non-linear functions by building separate locally linear models. In [28] an algorithm was presented for finding locally optimal regression by applying suitable weighting of the data for each cluster.

For voice conversion, it is important that the conversion function is continuous. Similarly to GMM-based techniques presented in Section II, smooth transitions as the function of the source vector can be accomplished by calculating the posterior probabilities (4) and setting the global prediction $\hat{\mathbf{y}}_t$ equal to the weighted sum of local predictions $\hat{\mathbf{y}}_{n,t}$:

$$\hat{\mathbf{y}}_t = \sum_{n=1}^{M} \omega_{n,t} \hat{\mathbf{y}}_{n,t}. \tag{11}$$

With the PLS regression model (10) for each local prediction, the above leads to the regression model

$$\mathbf{y}_t = \sum_{n=1}^{M} \omega_{n,t} \boldsymbol{\beta}_n \mathbf{x}_t + \mathbf{e}_t, \tag{12}$$

where $\boldsymbol{\beta}_n$ is the transform of cluster n.

Minimizing the local errors separately is not guaranteed to minimize the global error $\mathbf{e}_t$. To overcome this, we propose a technique where all the local transforms are estimated jointly in order to minimize the global error. We apply the SIMPLS on zero-mean variables, so that the vectors are centered first as follows. The target mean and locally weighted source and target means are calculated as $\boldsymbol{\mu}^y = \sum_{t=1}^{T} \mathbf{y}_t$,

$$\boldsymbol{\mu}_n^x = \frac{\sum_{t=1}^{T} \omega_{n,t}\mathbf{x}_t}{\sum_{t'=1}^{T} \omega_{n,t'}}, \text{ and } \boldsymbol{\mu}_n^y = \frac{\sum_{t=1}^{T} \omega_{n,t}\mathbf{y}_t}{\sum_{t'=1}^{T} \omega_{n,t'}}.$$

Centered source vectors are defined as $\tilde{\mathbf{y}}_t = \mathbf{y}_t - \boldsymbol{\mu}_y$, and a centered source vector where weighted duplicates of the original source vectors are augmented is defined as

$$\tilde{\mathbf{x}}_t = \begin{bmatrix} \omega_{1,t}(\mathbf{x}_t - \boldsymbol{\mu}_1^x) \\ \omega_{2,t}(\mathbf{x}_t - \boldsymbol{\mu}_2^x) \\ \vdots \\ \omega_{M,t}(\mathbf{x}_t - \boldsymbol{\mu}_M^x) \end{bmatrix}. \tag{13}$$

For centered vectors, model (12) can be written as

$$\tilde{\mathbf{y}}_t = \boldsymbol{\beta}\tilde{\mathbf{x}}_t + \mathbf{e}_t, \tag{14}$$

where

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_M]$$

Since (14) has the same form as the original PLS regression model (10), $\boldsymbol{\beta}$ can be estimated using the standard PLS algorithms described in Section III-A.

The final prediction is obtained by adding the means of each cluster as

$$\hat{\mathbf{y}}_t = \boldsymbol{\beta}\tilde{\mathbf{x}}_t + \sum_{n=1}^{M} \omega_{n,t} \boldsymbol{\mu}_n^y. \tag{15}$$

Even though the above mixture-regression model does not have exactly similar speaker-independent latent variable equivalence as in Eqs. (8) and (9), it effectively prevents overfitting, while still having the capability of modeling the dependence between source variables. Furthermore, since the transforms $\boldsymbol{\beta}_n$, $n = 1, \ldots, M$ are estimated simultaneously for all the Gaussians, the method is able to take into account the interaction between the clusters. The performance of the method is analyzed in detail in Section V.

IV. Posterior probability smoothing

A single GMM component usually dominates each frame in typical VC data, i.e., for each frame there is only one high posterior probability $\omega_{n,t}$. This is at least due to the high dimensionality of the data. In this paper, we consider a realistic case of VC in which only a little training data, 10 parallel sentences are available. For such a small amount of data one can reliably estimate only a small number of local transforms.

Figure 3 illustrates the frame-wise maxima of the component posterior probabilities for 15000 test data frames, the frames sorted to ascending posterior probabilities. Different numbers of Gaussians are illustrated with different line types. It can be seen that in one third of the frames a single component is dominating, their posterior probability being

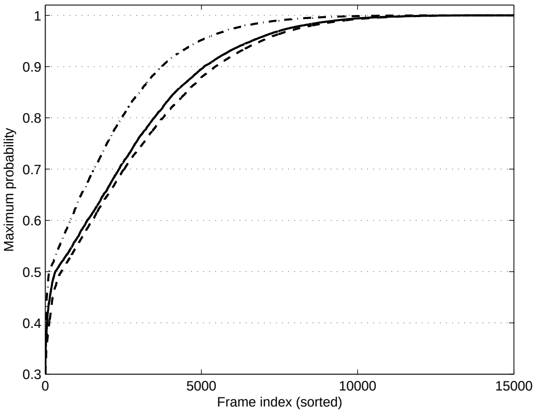

Fig. 3. *Frame-wise maximum GMM component posterior probabilities in a test set of source vectors sorted into ascending order. Different line types illustrates GMMs where different number of components were used (8 - dash-dotted line, 16 - solid line and 32 -dashed line).*

close to unity. In almost all the frames a single component has a posterior probability higher than 0.5. For example with 16 Gaussians, about 40 % of the data has a dominant component higher than 0.99. With training data that is more sparse (about 4200 frames), the percentage is even higher.

The "clustered" nature of posterior probabilities leads to rapid temporal changes from one component to another. This is illustrated in Figure 4 where the component posterior probabilities of a 8-component GMM are plotted with different line types over time. As can be seen, there is usually one dominant component for a short period of time and then it rapidly changes into another component. The temporal derivatives of the posterior probabilities are illustrated in Figure 5 with dashed line for all the components. The derivatives also show rapid temporal changes in the posterior probabilities.

Rapid changes in the posterior probabilities result in audible artefacts in the converted speech, since different transforms are used in each GMM component. This becomes prominent especially when the amount of training data is limited. To overcome this problem, the generated parameters were smoothed after the transforms in [24]. However, this easily produces overly smoothed features. Moreover, smoothing the features independently from each other may change their relationships, which causes problems when the features actually depend on each other. Postfiltering [29] is often used to improve the quality but it works in individual frames. Spectral trajectories [14] have been proposed to alleviate the problem, but the trajectories are difficult to estimate from a small amount of data.

To improve the perceptual quality of the converted samples with a small amount of data, we propose to smooth the component posterior probabilities before the transforms. Smoothing can be accomplished e.g. by a low-pass FIR filter and then normalizing the smoothed posterior probabilities in a frame so that they sum to unity.

In our system with 5 ms frame shift (200 Hz frame rate) we used a 10th order FIR lowpass filter having a cut-off frequency 10 Hz. The derivatives of the smoothed component posterior probabilities are illustrated in Figure 5 using solid lines. Smoothing does not change the fact that the data is sparse and clustered, but provides a smoother transition from one component to another.

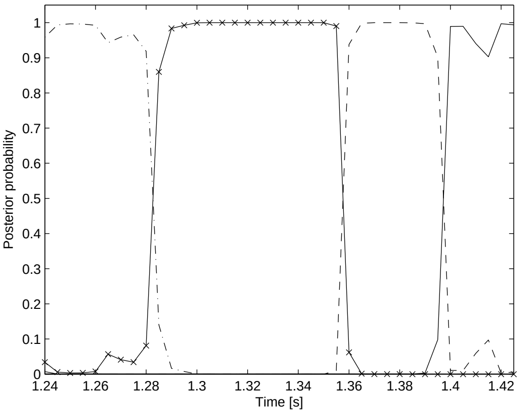

Fig. 4. *GMM component posterior probabilities for a speech segment, each component illustrated by a different line type. In most of the frames a single component is dominating, and the transitions between component are rapid.*

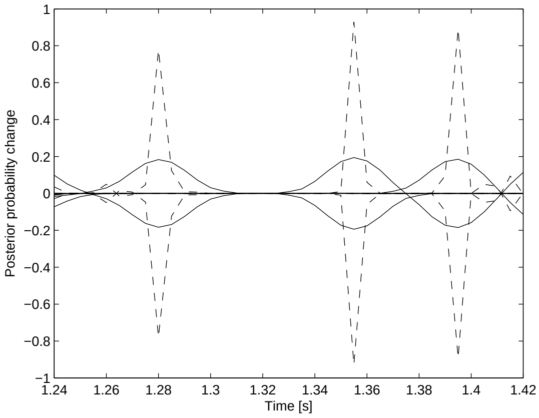

Fig. 5. *Temporal derivatives of the GMM component posterior probabilities in Figure 4 illustrated with dashed lines. The derivatives of smoothed posterior probabilities (solid lines) do not have as large changes.*

V. Experiments

Both objective and subjective results were carried out to evaluate the performance of the proposed methods. In the experiments, the proposed PLS model with GMM modeling explained in Section III-B (referred to as *PLS*) was evaluated against the joint-density GMM model explained in Section II-B (referred to as *JD*).

Full covariance matrices cannot be reliably estimated from a small amount of data. Therefore JD system used diagonal covariance matrices $\mathbf{\Sigma}_n^{xx}$, $\mathbf{\Sigma}_n^{xy}$, $\mathbf{\Sigma}_n^{yx}$, $\mathbf{\Sigma}_n^{yy}$ in the GMM estimation and feature transformation (Eq. (7)). PLS system used diagonal-covariance source GMM (Eq. (2)). Objective results were also calculated for multivariate regression based on a single transform.

JD was evaluated with 8, 16, and 32 GMM components, and because the model with 16 components was found to produce the smallest error, it was chosen as the baseline approach. The number of components in the source GMM of PLS was 8.

A. Acoustic data

The publicly available CMU Arctic database [30] sampled at 16 kHz was used for evaluation. We conducted tests for four speaker pairs: male-to-male (M-M), male-to-female (M-F), female-to-male (F-M) and female-to-female (F-F). The analysis-synthesis system STRAIGHT [31] was used for extracting F_0 and the spectral envelope at 5 ms steps. The spectral envelope was represented with 24-order MGCs [29] resulting in 25 cepstral parameters. The first term describing the energy was not used and in the sample generation it was copied from the source. The excitation was formed using either white noise or impulses, and the voicing decisions were directly copied from the source to the target. F_0 was converted by transforming the mean and variance in a logarithmic scale. Temporal differences were not modeled.

For each speaker pair, a source GMM and a JD GMM were built based on data from 10 sentences that were aligned with dynamic time warping (DTW). Some parts of the aligned training data were discarded, which was found to improve the objective quality in [32]. The training data selection process was automated so that silent frames were discarded based on an energy threshold and frames with voiced-unvoiced mismatch were omitted.

Objective results were calculated for the test data that had gone through a similar selection process as the training data. For example, comparing silent frames to silent frames is not meaningful. The testing data consisted of 35000 frames and did not include data from the training sentences.

B. Objective results

The Mel-cepstral distortion between the converted target and the original target was calculated as in [14] as

$$\mathrm{sd}_{\mathrm{mel}}[\mathrm{dB}] = \frac{10}{\ln 10}\sqrt{2\sum_{i=1}^{24}(c_i - \hat{c}_i)^2} \tag{16}$$

where c_i is the original target and $\hat{c}_i$ is the converted target of the ith MGC.

Figure 6 illustrates the average Mel-cepstral distortion for M-M, F-F, F-M, and M-F conversion, respectively. As can be seen from the panels, in all speaker pairs except for the female-to-female conversion, PLS with a suitable number of components (20-40) can yield a lower error than JD or a single full transform. Using a too low or too large number of components leads to worse results than the reference methods. The cepstral distortion between the original source and the target were 8.2 dB, 7.0 dB, 9.9 dB, and 9.3 dB for the M-M, F-F, F-M, and M-F conversions, respectively.

C. Subjective results

Preference tests were carried out concerning the speech quality and identity. Examples of the test samples are available at http://www.cs.tut.fi/sgn/arg/IEEE_VC/pls.html. Ten naive listeners participated in all the tests. The number of PLS components was set to 40, which was found to be a suitable number in the initial experiments with development data. All the PLS samples also used the proposed posterior probability smoothing. The samples produced by all the tested methods were postfiltered ($\beta = 0.4$) (see [29] for detailed description on postfiltering) and the samples were scaled to the same playback level.

In the quality test, we conducted two comparisons. In the first test (Test 1), PLS was tested against JD with posterior probability smoothing (referred to as JD-S). In the second test (Test 2), PLS was tested against JD without posterior probability smoothing (JD-NS). In both cases, 16 test sentence pairs from all the speaker pairs were evaluated. For each pair of samples produced by the two tested methods, the listeners were asked to choose the one with better quality.

In the identity test (Test 3), both systems (PLS and JD) used the proposed posterior probability smoothing. The subjects listened to the original target and were asked which sample, A or B, was closer to the target. The target sample was analyzed and synthesized similarly as the converted samples, i.e. using simple excitation and spectrum modeling with MGCs. 16 test sentences were evaluated for each speaker pair.

The average quality and identity preference results with 95 % confidence intervals for all speaker pairs are shown in Figure 7. A more detailed information about the votes given for each speaker pair are shown in Figure 8, Figure 9 and Figure 10 for Test 1 (quality), Test 2 (quality) and Test 3 (identity), respectively.

D. Analysis of the results

Both the objective analysis and the listening test results indicate a similar preference order for the compared systems. According to the preference test, PLS produces better quality than the smoothed JD-GMM. The only exception is the female-to-female transformation where the systems were rated equal within the confidence intervals. This is also depicted in the objective results, where female-to-female transformation with PLS did not reach a lower error with any number of PLS components compared to JD. In inter-gender transformation, as well as in male-to-male transformation, PLS is more often preferred. Compared to the other conversion pairs, the female speakers sound rather similar, and transforming each feature independently from each other by the JD method can perform equally well in comparison with PLS. Posterior probability smoothing improves the subjective quality of all the speaker pairs.

In the identity test, PLS was preferred more often in all cases except for the equal preference in female-to-female transformation. However, in all the samples, the compared samples were closer to each other than to the desired target because both of the methods are based on some type of linear transforms and similar parameterization.

VI. Discussion and future work

Modeling the conversion function as a weighted sum of local linear regression models gives a chance to approximate

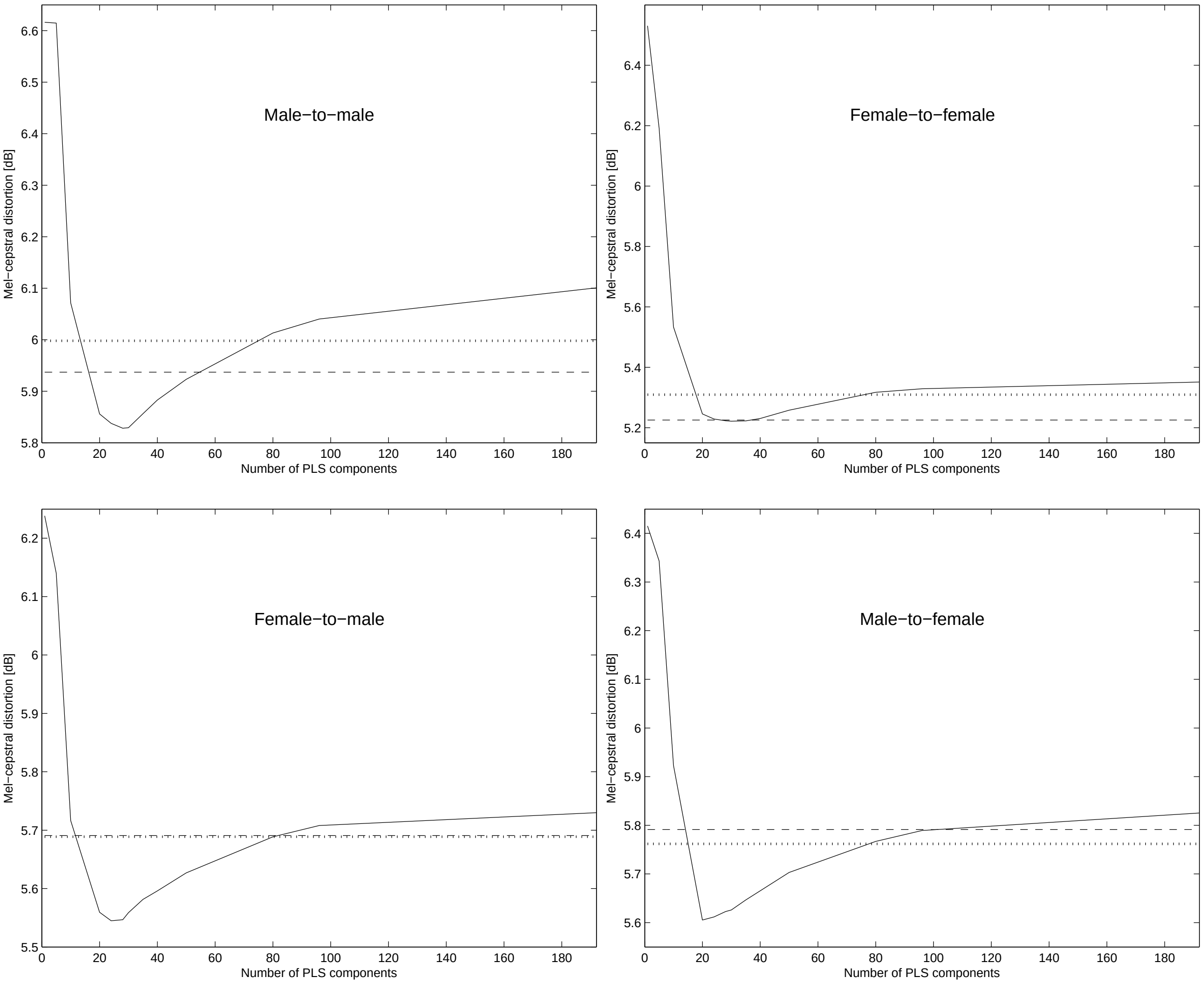

Fig. 6. *Mel-cepstral distortion of male-to-male (top left panel), female-to-female (top right panel), female-to-male (bottom left panel), and male-to-female (bottom right panel) transformations as a function of number of latent components. The dashed lines represent the error for the baseline JD method, the solid line represents the error of the proposed PLS method with different amount of latent components. The dotted line represents the error of multivariate regression with a single full transform matrix.*

non-linear models, but it is likely that the assumption of local linear dependence between the source and target features also limits the performance of all the methods discussed in this paper. To overcome the linearity restriction, non-linear PLS algorithms have been proposed, e.g. [33]. Detailed modeling is only possible when there is a large amount of data available which is, however, against the ultimate idea of voice conversion.

A source GMM adaptation in cases where a source GMM is built on a larger amount of data or model adaptation in HMM-based speech synthesis could also benefit from using PLS, at least when the source speaker or the master text-to-speech voice is significantly different from the voice to be adapted. Because of that, average-voice models perform better in adaptation [7]. When there is only a small amount of data available and no direct or well-established relationship between the source and the target parameters, simple models may fail and complex models tend to overtrain, while PLS offers a reasonable choice for balancing between these.

Even though we have used objective measurements to complement the listening test results, it should be emphasized that in voice conversion, the quality and identity cannot be reliably evaluated with objective measures. The objective results are based on aligned data which may not fully capture the true relationship between the source and the target. There can also be audible discontinuities, e.g. clicks, that may not affect the average objective measurements at all. In addition, the perceptual severity of the errors depend on the type of the sound (vowel, plosive, etc), a phenomenon that is hard to model objectively. Finally, it is possible that objective experiments could sometimes indicate higher distortion for samples that actually have better quality in a listening test.

VII. Conclusions

We have proposed a voice conversion method which combines PLS with GMMs. The method effectively prevents overfitting, while retaining the ability to model dependencies between features. We also proposed a method to improve the quality of GMM-based mapping by low-pass filtering the GMM component posterior probabilities.

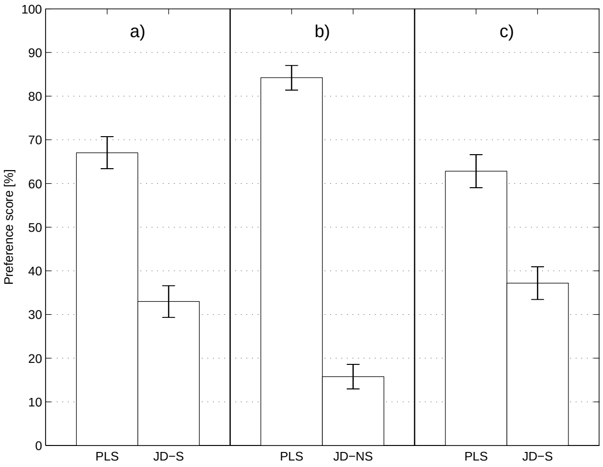

Fig. 7. *The overall results of the preference test with 95 % confidence intervals of the proposed PLS system in terms of a) speech quality against the smoothed baseline (JD-S) and b) speech quality against the non-smoothed baseline (JD-NS), and c) speech identity against the smoothed baseline (JD-S).*

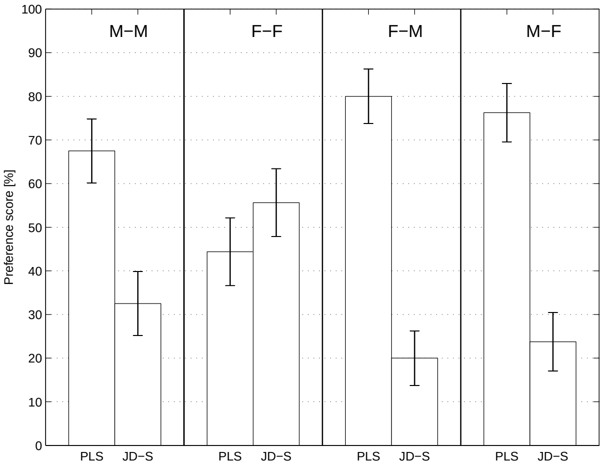

Fig. 8. *The overall results of the quality preference test for each speaker pair when evaluating the proposed PLS system against the smoothed baseline (JD-S) with 95 % of confidence intervals.*

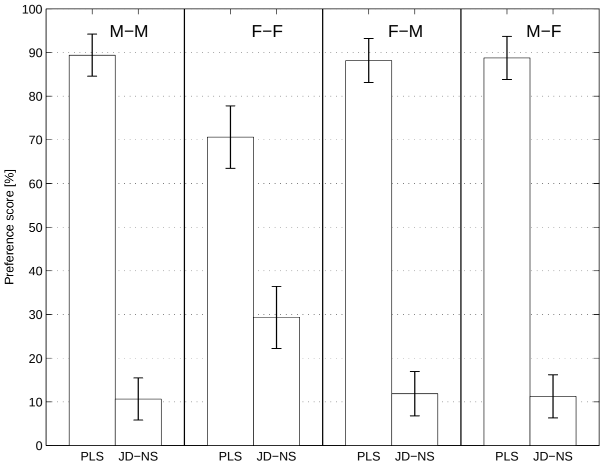

Fig. 9. *The results of the quality preference test for each speaker pair when evaluating the proposed PLS system against the baseline (JD-NS) with 95 % of confidence intervals.*

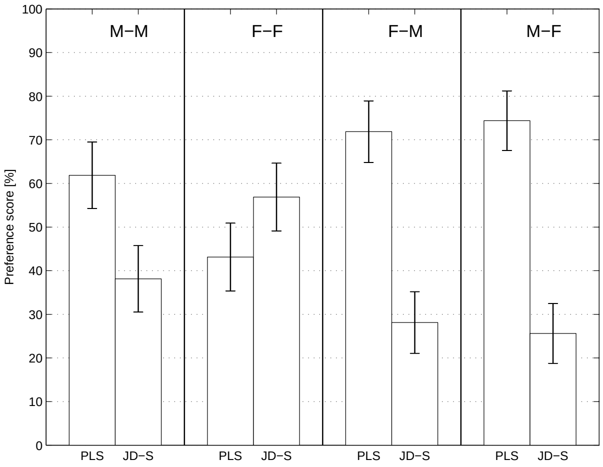

Fig. 10. *The results of the identity preference test for each speaker pair when evaluating the proposed PLS system against the smoothed baseline (JD-S) with 95 % of confidence intervals.*

Experimental results show that the proposed methods enable a better conversion quality than the baseline methods. In the cases where the difference between the source and the target speaker is large the proposed methods achieve a clearly better quality.

References

[1] K.-Y. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proc. of ICASSP*, vol. 3, 2000, pp. 1843–1846 vol.3.

[2] C.-C. Hsia, C.-H. Wu, and J.-Q. Wu, "Conversion function clustering and selection using linguistic and spectral information for emotional voice conversion," *Computers, IEEE Transactions on*, vol. 56, no. 9, pp. 1245–1254, Sept. 2007.

[3] A. Mouchtaris, J. Van der Spiegel, P. Mueller, and P. Tsakalides, "A spectral conversion approach to single-channel speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1180–1193, May 2007.

[4] S. Möller, *Assessment and Prediction of Speech Quality in Telecommunications*. Kluwer Academic Publisher, 2000.

[5] D. Sündermann, H. Höge, A. Bonafonte, H. Ney, and J. Hirschberg, "Text-independent cross-language voice conversion," in *Proc. of INTERSPEECH*, September 2006, pp. 2262–2265.

[6] J. Yamagishi, K. Ogata, Y. Nakano, J. Isogai, and T. Kobayashi, "HSMM-based model adaptation algorithms for average-voice-based speech synthesis," in *Proc. of ICASSP*, vol. I, Toulouse, May 2006, pp. 77–80.

[7] 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," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[8] T. Toda, Y. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," in *Proc. of ICASSP*, vol. 4, April 2007, pp. IV–1249–IV–1252.

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

[10] A. Mouchtaris, J. Van der Spiegel, and P. Mueller, "Nonparallel training for voice conversion based on a parameter adaptation approach," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 952–963, May 2006.

[11] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. on Speech and Audio Processing*, vol. 6(2), pp. 131–142, March 1998.

[12] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. of ICASSP*, vol. 1, 1998, pp. 285–288.

[13] O. Turk and L. Arslan, "Robust processing techniques for voice conversion," *Computer Speech and Language*, vol. 4(20), pp. 441–467, October 2006.

[14] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[15] D. Chapell and J. Hansen, "Speaker-specific pitch contour modelling and modification," in *Proc. of ICASSP*, Seattle, May 1998, pp. 885–888.

[16] B. Gillet and S. King, "Transforming F0 contours," in *Proc. of EUROSPEECH*, Geneve, September 2003, pp. 101–104.

[17] E. Helander and J. Nurminen, "A novel method for prosody prediction in voice conversion," in *Proc. of ICASSP*, vol. 4, April 2007, pp. IV–509–IV–512.

[18] M. Abe, S. Nakamura, K.Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. of ICASSP*, New York, April 1988, pp. 565–568.

[19] D. Erro, T. Polyakova, and A. Moreno, "On combining statistical methods and frequency warping for high-quality voice conversion," in *Proc. of ICASSP*, April 2008, pp. 4665–4668.

[20] Z. Shuang, F. Meng, and Y. Qin, "Voice conversion by combining frequency warping with unit selection," in *Proc. of ICASSP*, April 2008, pp. 4661–4664.

[21] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Communication*, no. 4, pp. 1–58, 1992.

[22] L. Mesbashi, V. Barreaud, and O. Boeffard, "Comparing GMM-based speech transformation systems," in *Proc. of INTERSPEECH*, 2007, pp. 1989–1456.

[23] Y. Uto, Y. Nankaku, T. Toda, A. Lee, and K. Tokuda, "Voice conversion based on mixtures of factor analyzers," in *Proc. of INTERSPEECH*, Pittsburgh, USA, September 2006, pp. 2278–2281.

[24] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice conversion with smoothed GMM and MAP adaptation," in *Proc. of EUROSPEECH*, 2003, pp. 2413–2416.

[25] D. Reynolds, "Experimental evaluation of features for robust speaker identification," *IEEE Trans. on Speech and Audio Processing*, vol. 2(4), pp. 639–643, 1994.

[26] H. Valbret, E. Moulines, and J. Tubach, "Voice transformation using PSOLA technique," in *Proc. of ICASSP*, vol. 1, Mar 1992, pp. 145–148.

[27] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, March 1993.

[28] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[29] K. Koishida, K. Tokuda, T. Kobayashi, and S. Imai, "CELP coding based on mel-cepstral analysis," in *Proc. of ICASSP*, vol. 1, May 1995, pp. 33–36.

[30] J. Kominek and A. Black, "CMU ARCTIC databases for speech synthesis," Carnegie Mellon University, Tech. Rep., 2003.

[31] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and a instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[32] E. Helander, J. Schwarz, J. Nurminen, H. Silén, and M. Gabbouj, "On the impact of alignment on voice conversion performance," in *Proc. of INTERSPEECH*, September 2008, pp. 1453–1456.

[33] G. Baffi, E. B. Martin, and A. J. Morris, "Non-linear projection to latent structures revisited," *Computers and Chemical Engineering*, vol. 23, no. 9, pp. 1293 – 1307, 1999.

Elina Helander received her M.S. degree in information technology in 2004 from Tampere University of Technology (TUT), Tampere, Finland. She is currently working as a researcher at the Institute of Signal Processing in TUT and pursuing towards the Ph.D. degree. Her research interests include voice conversion and modification, prosody modeling, and statistical speech synthesis.

Tuomas Virtanen received the M.Sc. and Doctor of Science degrees in information technology from the Tampere University of Technology (TUT), Finland, in 2001 and 2006, respectively. He is currently working as a senior researcher at TUT Department of Signal Processing. He has also been working as a research associate at Cambridge University Engineering Department, UK. His research interests include content analysis of audio signals, sound source separation, noise-robust automatic speech recognition, and machine learning.

Jani Nurminen received his M.Sc. degree from Dept. of Information Technology, Tampere University of Technology (TUT), Finland, in 2001. He has worked on speech related technologies since 1999, first in TUT until 2002, and after that in Nokia. He has authored or co-authored about 40 publications, and has over 30 granted or pending patents. Currently, he is a Technology Manager with Nokia Devices R&D. His research interests include speech, audio and language processing, speech synthesis, data compression, and multimodal user interfaces.

Moncef Gabbouj is currently a Professor at the Department of Signal Processing at Tampere University of Technology, Tampere, Finland. He was Head of the Department during 2002-2007. Dr. Gabbouj was on sabbatical leave at the American University of Sharjah, UAE in 2007-2008. Dr. Gabbouj was Senior Research Fellow of the Academy of Finland during 2007-2008 and 1997-1998. In 2007-2008, he was visiting professor at the American University of Sharjah, UAE. Dr. Gabbouj is the co-founder and past CEO of SuviSoft Oy Ltd. His research interests include multimedia content-based analysis, indexing and retrieval; nonlinear signal and image processing and analysis; and video processing, coding and communications.

Dr. Gabbouj is a Honorary Guest Professor of Jilin University, China (2005-2010). Dr. Gabbouj served as Distinguished Lecturer for the IEEE Circuits and Systems Society in 2004-2005, and Past-Chairman of the IEEE-EURASIP NSIP (Nonlinear Signal and Image Processing) Board. He was chairman of the Algorithm Group of the EC COST 211quat. He served as associate editor of the IEEE Transactions on Image Processing, and was guest editor of Multimedia Tools and Applications, the European journal Applied Signal Processing. He is the past chairman of the IEEE Finland Section, the IEEE Circuits and Systems Society, Technical Committee on Digital Signal Processing, and the IEEE SP/CAS Finland Chapter. He was also Chairman of CBMI 2005, WIAMIS 2001 and the TPC Chair of ISCCSP 2006 and 2004, CBMI 2003, EUSIPCO 2000, NORSIG 1996 and the DSP track chair of the 1996 IEEE ISCAS. He is also member of EURASIP Advisory Board and past member of AdCom. He also served as Publication Chair and Publicity Chair of IEEE ICIP 2005 and IEEE ICASSP 2006, respectively. Dr. Gabbouj was the recipient of the 2005 Nokia Foundation Recognition Award and co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award from the 1994 Nordic Signal Processing Symposium. He is co-author of over 400 publications.