Mixture of Factor Analyzers Using Priors from Non-Parallel Speech for Voice Conversion

Mixture of Factor Analyzers Using Priors from Non-Parallel Speech for Voice Conversion

Zhizheng Wu, Tomi Kinnunen, *Member, IEEE*, Eng Siong Chng, *Senior Member, IEEE*, and Haizhou Li, *Senior Member, IEEE*

***Abstract*—A robust voice conversion function relies on a large amount of parallel training data, which is difficult to collect in practice. To tackle the sparse parallel training data problem in voice conversion, this paper describes a mixture of factor analyzers method which integrates prior knowledge from non-parallel speech into the training of conversion function. The experiments on CMU ARCTIC corpus show that the proposed method improves the quality and similarity of converted speech. With both objective and subjective evaluations, we show the proposed method outperforms the baseline GMM method.**

***Index Terms*—Voice conversion, prior knowledge, factor analysis, mixture of factor analyzers.**

I. Introduction

THE objective of voice conversion is to manipulate one speaker's (source) voice to sound like that of another (target) without changing the phonetic information. It involves two processes: training and run-time conversion. During the training process, a conversion function is estimated to establish the relationship between the source and target speech features. In the conversion process, the conversion function is used to convert source speaker's voice to that of the target speaker. Apparently the conversion function has a direct impact on the quality of the resulting speech.

Many statistical methods have been adopted to implement the conversion function, such as mapping codebooks [1], artificial neural networks [2], [3], Gaussian mixture model [4], [5], [6], and partial least squares regression [7]. The *joint density Gaussian mixture model* (JD-GMM) [4], [5], [6] is one of the most effective approaches. Unfortunately, it requires relatively large parallel training data to avoid over-fitting [8].

There have been reported work on speech [9] and speaker recognition [10] where researchers leverage on existing speech corpora from non-target speakers as the prior knowledge to improve their systems' performance. Following the same idea, eigenvoice-based conversion [11], and tensor representation of speaker space [12] are examples of similar successful attempts in voice conversion. However, these methods all require a large amount of parallel data which are difficult to collect in practical situations.

In speaker verification, the *joint factor analysis* (JFA) method [13] decomposes a supervector into speaker independent, speaker dependent and channel dependent components, each of which is represented by a low-dimensional set of factors. Inspired by such an idea, we argue that similar decomposition would be useful in voice conversion, where we would like to separate phonetic and speaker specific components of speech spectral vectors, and apply factor analysis on the speaker specific component. The speaker specific component can then be represented by a low-dimensional set of latent variables via the factor loadings. To cover the intended speaker space densely, we adopt mixture of factor analyzers (MFA) [14], which was previously used to refine covariance of JD-GMM in voice conversion [15].

The main contribution of this work is a new technique that estimates the phonetic component and factor loadings from non-parallel prior data. In this way, during the training process, we only estimate a low-dimensional set of speaker identity factors and a tied covariance matrix instead of a full conversion function from the source-target parallel utterances. Even though parallel utterances are still required for estimating the conversion function, the use of prior data allows us to obtain a reliable model from much fewer training samples than those required by conventional JD-GMM [5].

II. Baseline joint density Gaussian mixture model

The mainstream joint density Gaussian mixture model (JD-GMM) conversion method [4] is used as our baseline.

Given parallel training utterances from source X and target Y speakers, dynamic time warping (DTW) can be applied to obtain the aligned feature vector pairs: $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, \ldots, \mathbf{z}_T]$, where $\mathbf{z}_t = [\mathbf{x}_t^\top, \mathbf{y}_t^\top]^\top \in \mathcal{R}^{2d}$ and $\mathbf{x}_t \in \mathcal{R}^d$, $\mathbf{y}_t \in \mathcal{R}^d$. The joint probability density of $\mathbf{X}$ and $\mathbf{Y}$ is modeled by a GMM as follows:

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Z}) = \sum_{k=1}^{K} \pi_k^{(z)} \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}_k^{(z)}), \qquad (1)$$

where $\boldsymbol{\mu}_k^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_k^{(x)} \\ \boldsymbol{\mu}_k^{(y)} \end{bmatrix}$ and $\boldsymbol{\Sigma}_k^{(z)} = \begin{bmatrix} \boldsymbol{\Sigma}_k^{(xx)} & \boldsymbol{\Sigma}_k^{(xy)} \\ \boldsymbol{\Sigma}_k^{(yx)} & \boldsymbol{\Sigma}_k^{(yy)} \end{bmatrix}$ are the mean vector and covariance matrix, respectively. Given the component k, $\pi_k^{(z)}$ is its prior probability with $\sum_{k=1}^{K} \pi_k^{(z)} = 1$. In the training phase, the GMM parameters $\lambda^{(z)} = \{\pi_k^{(z)}, \boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}_k^{(z)} | k = 1, 2, \ldots, K\}$ are estimated using the expectation maximization (EM) algorithm.

Z. Wu and E. S. Chng are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, and also with Temasek Lab@NTU, Nanyang Technological University, Singapore. (e-mail: wuzz@ntu.edu.sg; aseschng@ntu.edu.sg).

T. Kinnunen is with the School of Computing, University of Eastern Finland, Joensuu, Finland. (e-mail:tkinnu@cs.joensuu.fi). The work was supported by the Academy of Finland (project 132129 and 253120).

H. Li is with the Institute for Infocomm Research, Singapore 138632, University of New South Wales, Sydney, NSW 2052, Australia, and also with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hli@i2r.a-star.edu.sg).

In the conversion process, given a source speech feature vector $\mathbf{x}$, the joint density model is used for predicting the target speaker's feature vector $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$, where the conversion function $\mathcal{F}(.)$ is given as follows:

$$\mathcal{F}(\mathbf{x}) = \sum_{k=1}^{K} p_k(\mathbf{x})(\boldsymbol{\mu}_k^{(y)} + \boldsymbol{\Sigma}_k^{(yx)}(\boldsymbol{\Sigma}_k^{(xx)})^{-1}(\mathbf{x} - \boldsymbol{\mu}_k^{(x)})). \quad (2)$$

Here $p_k(\mathbf{x}) = \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^{(x)}, \boldsymbol{\Sigma}_k^{(xx)}) / \sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l^{(x)}, \boldsymbol{\Sigma}_l^{(xx)})$ is the posterior probability of source vector $\mathbf{x}$ belonging to the k-th Gaussian component.

III. Mixture of factor analyzers

In JD-GMM, we need to estimate many Gaussian components $\lambda^{(z)} = \{\pi_k^{(z)}, \boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}_k^{(z)} | k = 1, 2, \ldots, K\}$ from a large parallel training corpus for a reliable performance. To overcome this, we propose to use non-parallel prior data to estimate some speaker-independent parameters in advance, which are needed by the conversion.

Given a spectral vector, we assume that it consists of phonetic and speaker specific components, which are statistically independent. We further assume that the speaker specific component can be represented by a low-dimensional speaker identity vector (SIV) via a low-rank factor loading matrix. We use *factor analysis* model to represent this idea:

$$\mathbf{o} = \boldsymbol{\mu} + \mathbf{T}\mathbf{w} + \epsilon, \quad (3)$$

where $\mathbf{o} \in \mathcal{R}^d$ is an observed d-dimensional spectral vector, $\boldsymbol{\mu} \in \mathcal{R}^d$ is the speaker-independent phonetic component, $\mathbf{T}\mathbf{w}$ is the speaker specific component in which $\mathbf{w} \in \mathcal{R}^{m \times 1}$ is the latent SIV and $\mathbf{T} \in \mathcal{R}^{d \times m}$ is the factor loading matrix. ϵ is the noise term.

Factor analysis is a linear single-Gaussian latent variable model. However, as speech data can *not* be well represented by a single Gaussian, we adopt the mixture of factor analyzers (MFA) model [14]. The likelihood function of the non-parallel prior data $\mathbf{O} = [\mathbf{o}_1^{(1)}, \mathbf{o}_2^{(1)}, \ldots, \mathbf{o}_{N_1}^{(1)}, \ldots, \mathbf{o}_{N_S}^{(S)}]^\top$ for the model $\lambda^{(\mathrm{MFA})} = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{T}_k, \boldsymbol{\Sigma}_k | k = 1, 2, \ldots, K\}$ is :

$$\begin{aligned} P(\mathbf{O}, \mathbf{w}|\lambda^{(\mathrm{MFA})}) &= P(\mathbf{O}|\mathbf{w}, \lambda^{(\mathrm{MFA})}) P(\mathbf{w}) \\ &= \prod_{s=1}^{S} P(\mathbf{O}^{(s)}|\mathbf{w}_s, \lambda^{(\mathrm{MFA})}) P(\mathbf{w}_s) \quad (4) \end{aligned}$$

$$P(\mathbf{O}^{(s)}|\mathbf{w}_s, \lambda^{(\mathrm{MFA})}) = \prod_{n=1}^{N_s} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{o}_n^{(s)}|\boldsymbol{\mu}_k + \mathbf{T}_k \mathbf{w}_s, \boldsymbol{\Sigma}_k). \quad (5)$$

$$P(\mathbf{w}_s) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (6)$$

where $\mathcal{N}$ is the Gaussian function, S represents the number of speakers, and N_s represents the number of frames from the s-th speaker, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathcal{R}^d$ represent speaker independent phonetic vectors, $\mathbf{w}_s \in \mathcal{R}^m$ is the SIV of speaker s, $\mathbf{T}_k \in \mathcal{R}^{d \times m}$ is the factor loadings of the k-th factor analyzer component with prior probability π_k and $\sum_{k=1}^{K} \pi_k = 1$.

The proposed spectral conversion framework is presented in Fig. 1. In off-line process, we use non-parallel prior corpus

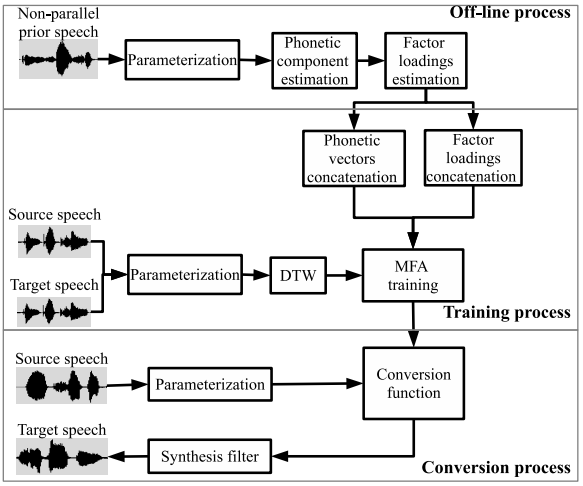


Fig. 1. Proposed spectral conversion system

to estimate the phonetic component $\boldsymbol{\mu}_k$ and factor loadings $\mathbf{T}_k$ in section 3.A and 3.B, respectively. Then we adopt $\boldsymbol{\mu}_k$ and $\mathbf{T}_k$ to jointly estimate the speaker identity vectors $\mathbf{w}^{(x)}$, $\mathbf{w}^{(y)}$ for source and target in section 3.C, and finally derive the conversion function, which is similar as equation (2).

A. Speaker-independent phonetic vectors estimation

In theory, we could estimate all the parameters $\lambda^{(\mathrm{MFA})}$ at the same time as in [14]. To benefit from a large speaker independent database [13] and ensure that the phonetic vectors are not affected by the speaker-specific component when estimating the factor loadings, we use pre-trained GMM to represent the phonetic space. While a Gaussian component may not correspond to a phonetic unit exactly, we assume that a mixture of Gaussian components cover the whole phonetic space. In this way, the likelihood function for the phonetic GMM $\lambda^{(\mathrm{phonetic})} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | k = 1, 2, \ldots, K\}$ is written as,

$$P(\mathbf{O}|\lambda^{(\mathrm{phonetic})}) = \prod_{s=1}^{S} \prod_{n=1}^{N_s} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{o}_n^{(s)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (7)$$

where $\boldsymbol{\mu}_k \in \mathcal{R}^d$ is an estimated phonetic vector, and $\boldsymbol{\Sigma}_k \in \mathcal{R}^{d \times d}$ is the covariance matrix. EM algorithm is used to estimate the parameters $\lambda^{(\mathrm{phonetic})}$. The π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ in (5) are replaced by that in (7), and π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are fixed when estimating the factor loading matrices $\mathbf{T}_k$.

B. Speaker-independent factor loadings estimation

Given $\lambda^{(\mathrm{phonetic})}$, we use EM algorithm to estimate the factor loading matrices $\mathbf{T}_k$ in (4), as there are latent variables $\mathbf{w}$. The E-step and M-step are written as follows:

1) E-step: Calculate the occupation probability $\gamma_n^{(s)}(k)$ and the expectation of latent variable $\mathbf{w}_s$:

$$\gamma_n^{(s)}(k) = \frac{\pi_k P(\mathbf{o}_n^{(s)}|\mathbf{w}_s, \mathbf{T}_k', \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l P(\mathbf{o}_n^{(s)}|\mathbf{w}_s, \mathbf{T}_l', \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \quad (8)$$

$$\mathbb{E}[\mathbf{w}_s] = \mathbf{F}^{-1} \cdot \sum_{n=1}^{N_s} \sum_{k=1}^{K} \gamma_n^{(s)}(k) \mathbf{T}_k'^{\top} \boldsymbol{\Sigma}_k^{-1}(\mathbf{o}_n^{(s)} - \boldsymbol{\mu}_k) \quad (9)$$

$$\mathbb{E}[\mathbf{w}_s\mathbf{w}_s^\top] = \mathbf{F}^{-1} + \mathbb{E}[\mathbf{w}_s]\mathbb{E}[\mathbf{w}_s]^\top, \tag{10}$$

where $\mathbf{F} = \mathbf{I} + \sum_{n=1}^{N_s}\sum_{k=1}^{K}\gamma_n^{(s)}(k)\mathbf{T}_k^{'\top}\Sigma_k^{-1}\mathbf{T}_k^{'}$ and $\mathbf{T}_k^{'}$ are the factor loading matrices estimated in previous M-step.

2) M-step: Estimate the new factor loading matrices $\mathbf{T}_k$:

$$\mathbf{T}_k = \frac{\sum_{s=1}^{S}\sum_{n=1}^{N_s}\gamma_n^{(s)}(k)\cdot\mathbb{E}[\mathbf{w}_s](\mathbf{o}_n^{(s)} - \boldsymbol{\mu}_k)}{\sum_{s=1}^{S}\sum_{n=1}^{N_s}\gamma_n^{(s)}(k)\mathbb{E}[\mathbf{w}_s\mathbf{w}_s^\top]} \tag{11}$$

We run 10 EM iterations for estimating the factor loading matrices $\mathbf{T}_k$, which are randomly initialized at the beginning.

C. Voice conversion using mixture of factor analyzers

Now that we have estimated the factor loadings and phonetic vectors from non-parallel prior corpora, we can estimate the conversion function from parallel data $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, \ldots, \mathbf{z}_T]$, where $\mathbf{z}_t = [\mathbf{x}_t^\top, \mathbf{y}_t^\top]^\top \in \mathcal{R}^{2d}$, we concatenate the phonetic vectors as $\boldsymbol{\mu}_k^{(z)} = [\boldsymbol{\mu}_k^\top, \boldsymbol{\mu}_k^\top]^\top \in \mathcal{R}^{2d}$ and the factor loadings as $\mathbf{A}_k = \begin{bmatrix}\mathbf{T}_k & \mathbf{0}\\ \mathbf{0} & \mathbf{T}_k\end{bmatrix} \in \mathcal{R}^{2d\times 2m}$. We note that the two $\boldsymbol{\mu}_k$ in $\boldsymbol{\mu}_k^{(z)}$ are identical and the two $\mathbf{T}_k$ in $\mathbf{A}_k$ are also identical. This concatenation will not change the phonetic mapping when training conversion function. Thus the joint distribution for the parallel data is written as follows.

$$P(\mathbf{Z}|\mathbf{w}^{(z)}) = \sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k^{(z)} + \mathbf{A}_k\mathbf{w}^{(z)}, \boldsymbol{\Sigma}^{(z)}). \tag{12}$$

Here $\mathbf{w}^{(z)} = [\mathbf{w}^{(x)\top}, \mathbf{w}^{(y)\top}]^\top \in \mathcal{R}^{2m\times 1}$ is the joint speaker identity vector where $\mathbf{w}^{(x)} \in \mathcal{R}^{m\times 1}$ is for source speaker and $\mathbf{w}^{(y)} \in \mathcal{R}^{m\times 1}$ is for target speaker, and $\boldsymbol{\Sigma}^{(z)} = \begin{bmatrix}\boldsymbol{\Sigma}^{(xx)} & \boldsymbol{\Sigma}^{(xy)}\\ \boldsymbol{\Sigma}^{(yx)} & \boldsymbol{\Sigma}^{(yy)}\end{bmatrix} \in \mathcal{R}^{2d\times 2d}$ is a covariance matrix. A full covariance matrix consists of a large number of free parameters which need to be estimated. To circumvent this data sparseness and avoid numerical problem, we use a tied covariance matrix shared by all the Gaussians in implementation. We dub our method as *tied mixture of factor analyzers* (TMFA). The benefit of using factor loadings is that when estimating speaker specific components, we only need to estimate a low-dimensional SIV with less training data, as the factor loadings are estimated in advance. Similar as that for equation (4), EM algorithm can be adopted to estimate $\mathbf{w}^{(z)}$ and $\boldsymbol{\Sigma}^{(z)}$ under the maximum likelihood criterion:

1) E-step: calculate the occupation probability $p_k(\mathbf{z_t})$ and joint speaker identity vector $\mathbf{w}^{(z)}$:

$$p_k(\mathbf{z_t}) = \frac{\pi_k P(\mathbf{z}_t|\mathbf{w}^{(z)}, \mathbf{A}_k, \boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}^{(z)})}{\sum_{l=1}^{K}\pi_l P(\mathbf{z}_t|\mathbf{w}^{(z)}, \mathbf{A}_l, \boldsymbol{\mu}_l^{(z)}, \boldsymbol{\Sigma}^{(z)})} \tag{13}$$

$$\mathbf{w}^{(z)} = \frac{\sum_{t=1}^{T}\sum_{k=1}^{K}p_k(\mathbf{z}_t)\mathbf{A}_k^\top\boldsymbol{\Sigma}^{(z)^{-1}}(\mathbf{z}_t - \boldsymbol{\mu}_k^{(z)})}{\mathbf{I} + \sum_{t=1}^{T}\sum_{k=1}^{K}p_k(\mathbf{z}_t)\mathbf{A}_k^\top\boldsymbol{\Sigma}^{(z)^{-1}}\mathbf{A}_k^\top} \tag{14}$$

2) M-step: estimate new tied covariance matrix $\boldsymbol{\Sigma}^{(z)}$:

$$\boldsymbol{\Sigma}^{(z)} = \frac{\sum_{t=1}^{T}\sum_{k=1}^{K}p_k(\mathbf{z_t})\mathbf{v}\mathbf{v}^\top}{\sum_{t=1}^{T}\sum_{k=1}^{K}p_k(\mathbf{z_t})}, \tag{15}$$

where $\mathbf{v} = \mathbf{z_t} - \boldsymbol{\mu}_k^{(z)} - \mathbf{A}_k\mathbf{w}^{(z)}$. In this EM algorithm, we initialize the tied covariance matrix with global covariance matrix and initialize $\mathbf{w}^{(z)}$ as zero vector. We run three EM iterations to estimate $\boldsymbol{\Sigma}^{(z)}$ and $\mathbf{w}^{(z)}$.

In the conversion process, given $\mathbf{x}$, the tied joint-density MFA model is adopted to predict the target feature vector $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$ as follows:

$$\mathcal{F}(\mathbf{x}) = \sum_{k=1}^{K}p_k(\mathbf{x})\cdot(\boldsymbol{\mu}_k + \mathbf{T}_k\mathbf{w}^{(y)} + \boldsymbol{\Sigma}^{(yx)}(\boldsymbol{\Sigma}^{(xx)})^{-1}(\mathbf{x} - \boldsymbol{\mu}_k - \mathbf{T}_k\mathbf{w}^{(x)}))$$

where $p_k(\mathbf{x})$ is the occupation probability of $\mathbf{x}$ belonging to the k-th factor analyzer.

IV. Experiments

We conduct conversion tests on CMU ARCTIC corpus for two speaker pairs: male-to-male (M-M, BDL-to-RMS) and female-to-female (F-F, SLT-to-CLB). We use 2 to 8 utterances of each speaker as the training data, and 50 utterances as testing data. Aurora 4 corpus, which has 83 speakers and each speaker has around 100 utterances (clean speech), is used as the prior data to estimate phonetic vectors and factor loadings.

The speech signal is sampled at 16kHz. Spectral envelope and fundamental frequency (F0) are extracted by STRAIGHT [16] at 5ms step, and the spectral envelope is parameterized as 25-order mel-cepstral coefficients (MCC), including the energy coefficient, which is not converted. Hence only 24-order coefficients are converted. F0 is converted by equalizing the mean and variance of the source and target speakers.

The following conversion methods are compared:

1) *GMM-full*: JD-GMM with full covariance matrices.
2) *GMM-cross*: JD-GMM with covariance matrices which have only diagonal and cross-covariance elements [15].
3) *TMFA-full*: TMFA with full covariance matrices.
4) *TMFA-cross*: TMFA with covariance matrices which only have diagonal and cross-covariance elements.

A. Objective evaluation

The mel-cepstral distortion (MCD) is used as the objective evaluation measure between a converted target frame and a original target frame [6]: $\mathrm{MCD[dB]} = \frac{10}{\log 10}\sqrt{2\sum_{d=1}^{24}(c_d - c_d^{(\mathrm{converted})})^2}$ where c_d and $c_d^{(\mathrm{converted})}$ are the d-th original target and converted MCCs, respectively. A lower MCD value indicates smaller distortion.

We first compare the conversion method using two training utterances. Fig. 2 shows the average MCD values of M-M and F-F spectral conversions as a function of the number of factors in TMFA. The baseline JD-GMM model has one Gaussian component, as it gives the lower MCD value than 2 or 4 Gaussian components. There are 128 Gaussians in TMFA with the number of factors in TMFA varying from 8 to 64. When more than 24 factors are used, TMFA gives much lower distortion than the baseline JD-GMM does. Another observation is that TMFA-cross always outperforms TMFA-full that suggests the latter suffers from over fitting.

We further train TMFA-cross and GMM-cross with a different amount of parallel training data. The number of factors is set to be $m = 44$, which is the median number between 24 and 64. The average MCD values of M-M and F-F conversion are

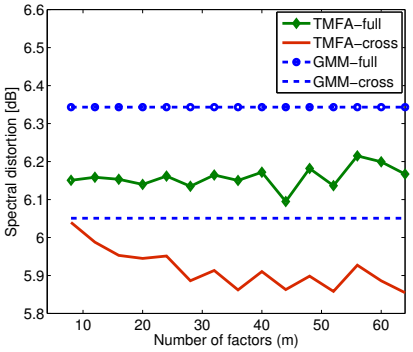
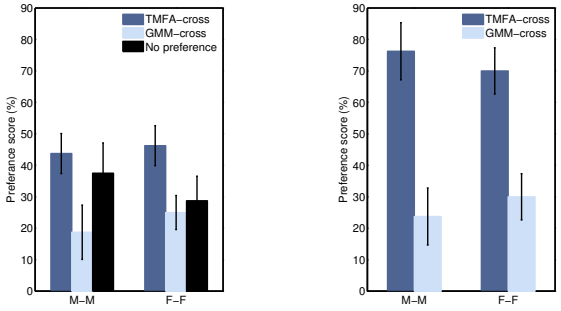


Fig. 2. Average mel-cepstral distortion of in terms of number of factors



(a) Similarity results (b) Overall quality results

Fig. 4. Subjective evaluation results with 95% confidence interval

presented in Fig. 3. TMFA outperforms JD-GMM when we have a limited amount of parallel training data, in particular, when the number of parallel utterances is less than 7. In general, TMFA model has fewer parameters and is more robust than JD-GMM due to the prior knowledge that it learnt from non-parallel prior data.

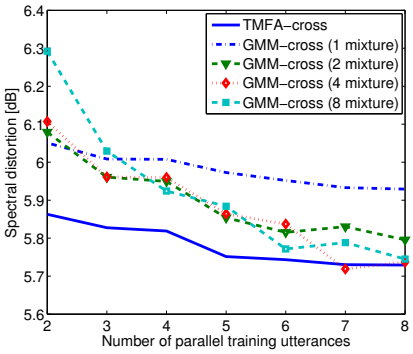


Fig. 3. Average MCD in terms of number of training utterances

B. Subjective evaluation

TMFA-cross with 44 factors is compared with GMM-cross with 1 mixture in the listening test. The number of training utterances is two. Similarity of the converted speech was first evaluated in an AB preference test. 8 subjects participated in the listening test. They were asked to listen to two converted speech (A and B), and the reference speech, and decide which converted speech sounded more similar to the reference speech by choosing one of the followings: 1) A is more similar; 2) B is more similar; 3) no preference. 10 sentences were evaluated for each speaker pair. The similarity preference results are shown in Fig. 4(a). We can see that TMFA technique consistently outperforms JD-GMM in both test cases.

The AB preference test was also conducted to evaluate the perceptual quality of the converted speech. Eight subjects listened to 10 sentence pairs for each speaker pair, and decided which converted speech they preferred. The quality preference test results are presented in Fig. 4(b). It shows that TMFA outperforms JD-GMM perceptually.

V. Conclusion

We proposed a voice conversion technique based on mixture of factor analyzers, by assuming that a speech spectral vector consists of independent phonetic and speaker specific components. We have shown that the prior knowledge from non-parallel data serves well in covering the feature space. With objective and subjective evaluations, we show our proposed method outperforms the conventional JD-GMM method.

References

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *ICASSP*, 1988.

[2] M. Narendranath, H. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, no. 2, pp. 207–216, 1995.

[3] S. Desai, E. Raghavendra, B. Yegnanarayana, A. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *ICASSP*, 2009.

[4] A. Kain and M. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, 1998.

[5] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *Speech and Audio Processing, IEEE Transactions on*, vol. 6, no. 2, pp. 131–142, 1998.

[6] 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, 2007.

[7] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 912–921, 2010.

[8] E. Helander, J. Nurminen, and M. Gabbouj, "LSF mapping for voice conversion with very small training sets," in *ICASSP*, 2008.

[9] R. Kuhn, J. 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, 2000.

[10] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

[11] T. Toda, Y. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," in *ICASSP*, 2007.

[12] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Proc. of Interspeech*, 2011.

[13] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[14] Z. Ghahramani and G. Hinton, "The EM algorithm for mixtures of factor analyzers," Technical Report CRG-TR-96-1, University of Toronto, Tech. Rep., 1996.

[15] Y. Uto, Y. Nankaku, T. Toda, A. Lee, and K. Tokuda, "Voice conversion based on mixtures of factor analyzers," in *Ninth International Conference on Spoken Language Processing*, 2006.

[16] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.