ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Implementation of Computationally Efficient Real-Time Voice Conversion

Tomoki Toda[1], *Takashi Muramatsu*[1], *Hideki Banno*[2]

[1]Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, JAPAN
[1]Graduate School of Science and Technology, Meijo University
Siogamaguchi 1-501, Tempaku-ku, Nagoya-shi, Aichi 468-5802, JAPAN
tomoki@is.naist.jp

Abstract

This paper presents an implementation of real-time processing of statistical voice conversion (VC) based on Gaussian mixture models (GMMs). To develop VC applications for enhancing our human-to-human speech communication, it is essential to implement real-time conversion processing. Moreover, it is useful to reduce computational complexity of the conversion processing for making VC applications available even in limited resources. In this paper, we propose a real-time VC method based on a low-delay conversion algorithm considering dynamic features and a global variance. Moreover, we also propose a computationally efficient VC method based on rapid source feature extraction and diagonalization of full covariance matrices. Some experimental results are presented to show that the proposed methods work reasonably well.

Index Terms: voice conversion, real-time processing, low-delay conversion, computational efficiency

1. Introduction

Statistical voice conversion (VC) is an effective technique for modifying acoustic parameters to convert non-linguistic information while keeping linguistic information unchanged. There are a lot of applications of this technique for enhancing our human-to-human speech communication beyond various constraints causing some barriers; *e.g.*, speaking-aid for handicapped people beyond physical constraints [1]. To develop such VC applications, it is essential to implement real-time conversion processing. A conversion method based on a Gaussian mixture model (GMM) [2] is a promising technique since it enables frame-by-frame conversion processing and no text transcription is necessary.

As one of the state-of-the-art GMM-based conversion methods, a trajectory-based conversion method has been proposed [3] but it does not basically run in real time. Towards real-time VC processing, a low-delay conversion algorithm to approximate the trajectory-based conversion process with a frame-by-frame conversion process has been proposed [4] inspired by a recursive parameter generation algorithm for speech synthesis based on hidden Markov model [5] and its another application for speech coding [6]. However, it does not consider a global variance (GV), which is helpful to significantly improve converted speech quality. Moreover, it will be required to reduce computational cost in the conversion process as much as possible to implement VC applications using only limited resources.

This paper presents an implementation method of computationally efficient real-time VC processing. The GV is implemented as postfiltering process to improve quality of converted speech. Moreover, to reduce computational complexity, the GMM is modified so as to accept rapidly extracted source features and approximate likelihood calculation.

2. Low-Delay Voice Conversion Based on Trajectory Estimation

2.1. Feature Extraction

As the source feature, the $D^{(x)}$-dimensional spectral segment feature vector $\boldsymbol{X}_t$ at frame t is extracted from a joint vector developed by concatenating spectral parameter vectors over several frames from $t-C$ to $t+C$ of the source voice as follows:

$$\boldsymbol{X}_t = \boldsymbol{E}\left[\boldsymbol{x}_{t-C}^\top, \cdots, \boldsymbol{x}_t^\top, \cdots, \boldsymbol{x}_{t+C}^\top\right]^\top + \boldsymbol{f}, \quad (1)$$

where $^\top$ denotes transposition of the vector. There are some options of a setting of the transformation matrix $\boldsymbol{E}$ and the bias vector $\boldsymbol{f}$; *e.g.*, using regression coefficients to calculate dynamic features or using eigenvectors to efficiently model the joint vector. This setting depends on each of VC applications.

As the target feature, a joint static and dynamic feature vector $\boldsymbol{Y}_t = \left[\boldsymbol{y}_t^\top, \Delta\boldsymbol{y}_t^\top\right]^\top$ is calculated at each frame, where $\boldsymbol{y}_t$ is a $D^{(y)}$-dimensional speech parameter vector of the target voice at frame t and $\Delta\boldsymbol{y}_t$ is its dynamic feature vector, which is calculated as

$$\Delta\boldsymbol{y}_t = \boldsymbol{y}_t - \boldsymbol{y}_{t-1}. \quad (2)$$

It depends on VC applications which speech parameter is used.

2.2. Training

The joint source and target feature vector $\left[\boldsymbol{X}_t^\top, \boldsymbol{Y}_t^\top\right]^\top$ is developed at each frame by performing time alignment between a time sequence of the source feature vectors and that of the target feature vectors in a training data set. Then, the joint probability density function (*p.d.f.*) of the source and target feature vectors is modeled with a GMM as follows:

$$P\left(\boldsymbol{X}_t, \boldsymbol{Y}_t | \boldsymbol{\lambda}^{(X,Y)}\right) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\left[\boldsymbol{X}_t^\top, \boldsymbol{Y}_t^\top\right]^\top; \boldsymbol{\mu}_m^{(X,Y)}, \boldsymbol{\Sigma}_m^{(X,Y)}\right), \quad (3)$$

where the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ is denoted as $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, the mixture component index is m, the total number of mixture components is M, and $\boldsymbol{\lambda}^{(X,Y)}$ denotes a parameter set of the GMM. The weight of the m^{th} mixture component is α_m. The mean vector $\boldsymbol{\mu}_m^{(X,Y)}$ and the covariance matrix $\boldsymbol{\Sigma}_m^{(X,Y)}$ of the m^{th} mixture component are respectively written as

$$\boldsymbol{\mu}_m^{(X,Y)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(X)} \\ \boldsymbol{\mu}_m^{(Y)} \end{bmatrix}, \quad \boldsymbol{\Sigma}_m^{(X,Y)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(XX)} & \boldsymbol{\Sigma}_m^{(XY)} \\ \boldsymbol{\Sigma}_m^{(YX)} & \boldsymbol{\Sigma}_m^{(YY)} \end{bmatrix}. \quad (4)$$

2.3. Conversion

Let $\boldsymbol{X} = \left[\boldsymbol{X}_1^\top, \cdots, \boldsymbol{X}_T^\top\right]^\top$ and $\boldsymbol{Y} = \left[\boldsymbol{Y}_1^\top, \cdots, \boldsymbol{Y}_T^\top\right]^\top$ be a time sequence vector of the source feature vectors and that of the target feature vectors, respectively. A time sequence vector of the converted static feature vectors $\hat{\boldsymbol{y}} = \left[\hat{\boldsymbol{y}}_1^\top, \cdots, \hat{\boldsymbol{y}}_T^\top\right]^\top$ is determined by maximizing the conditional *p.d.f.* of $\boldsymbol{Y}$ given $\boldsymbol{X}$ [3] as follows:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} P\left(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\lambda}^{(X,Y)}\right) \quad \text{subject to} \quad \boldsymbol{Y} = \boldsymbol{W}\boldsymbol{y}, \quad (5)$$

where $\boldsymbol{W}$ is the $2D^{(y)}T$-by-$D^{(y)}T$ matrix to extend a time sequence vector of the static feature vectors into that of the joint static and dynamic feature vectors [5]. In the low-delay conversion algorithm [4], the suboptimum mixture component sequence $\hat{\boldsymbol{m}} = \{\hat{m}_1, \cdots, \hat{m}_T\}$ is determined frame by frame as follows:

$$\hat{m}_t \quad = \quad \arg\max_m P(m|\boldsymbol{X}_t, \boldsymbol{\lambda}^{(X,Y)}). \quad (6)$$

The conditional *p.d.f.* of $\boldsymbol{Y}_t$ given $\boldsymbol{X}_t$ and the m^{th} mixture component at each frame is modeled by a Gaussian distribution, where its mean vector and covariance matrix are given by

$$\boldsymbol{\mu}_{m,t}^{(Y|X)} = \boldsymbol{\mu}_m^{(Y)} + \boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{\Sigma}_m^{(XX)^{-1}}\left(\boldsymbol{X}_t - \boldsymbol{\mu}_m^{(X)}\right), \quad (7)$$

$$\boldsymbol{\Sigma}_m^{(Y|X)} = \boldsymbol{\Sigma}_m^{(YY)} - \boldsymbol{\Sigma}_m^{(YX)}\boldsymbol{\Sigma}_m^{(XX)^{-1}}\boldsymbol{\Sigma}_m^{(XY)}, \quad (8)$$

respectively. Using only diagonal components of the covariance matrix $\boldsymbol{\Sigma}_m^{(Y|X)}$, each dimensional components of $\hat{\boldsymbol{y}}$ is separately determined. A $(L+1)$-by-$(L+1)$ state covariance matrix $\boldsymbol{P}_d^{(0)}$ and a $(L+1)$-dimensional state vector $\hat{\boldsymbol{y}}_d^{(0)}$ are initialized as the zero matrix and the zero vector, respectively. Then, they are recursively updated frame by frame as follows:

$$\boldsymbol{P}_d'^{(t-1)} = \boldsymbol{J}_L\boldsymbol{P}_d^{(t-1)}\boldsymbol{J}_L^\top + \text{diag}\left[\boldsymbol{0}_{1\times L}, \Sigma_{m_t,d}^{(y|X)}\right], \quad (9)$$

$$\hat{\boldsymbol{y}}_d'^{(t-1)} = \boldsymbol{J}_L\hat{\boldsymbol{y}}_d^{(t-1)} + \left[\boldsymbol{0}_{1\times L}, \mu_{m_t,t,d}^{(y|X)}\right]^\top, \quad (10)$$

$$\boldsymbol{P}_d^{(t)} = \left(\boldsymbol{I} - \boldsymbol{k}_d^{(t)}\boldsymbol{w}_L\right)\boldsymbol{P}_d'^{(t-1)}, \quad (11)$$

$$\hat{\boldsymbol{y}}_d^{(t)} = \hat{\boldsymbol{y}}_d'^{(t-1)} + \boldsymbol{k}_d^{(t)}\left(\mu_{m_t,t,d}^{(\Delta y|X)} - \boldsymbol{w}_L\hat{\boldsymbol{y}}_d'^{(t-1)}\right), \quad (12)$$

where the $(L+1)$-dimensional vector $\boldsymbol{k}_d^{(t)}$ is calculated as

$$\boldsymbol{k}_d^{(t)} = \boldsymbol{P}_d^{(t-1)}\boldsymbol{w}_t^\top\left(\Sigma_{m_t,d}^{(\Delta y|X)} + \boldsymbol{w}_L\boldsymbol{P}_d^{(t-1)}\boldsymbol{w}_L^\top\right)^{-1}, \quad (13)$$

and the $(L+1)$-dimensional row vector $\boldsymbol{w}_L$ and the $(L+1)$-by-$(L+1)$ matrix $\boldsymbol{J}_L$ are given by

$$\boldsymbol{w}_L = \left[\boldsymbol{0}_{1\times(L-1)}, -1, 1\right], \quad \boldsymbol{J}_L = \begin{bmatrix} 0 & \boldsymbol{I}_{L\times L} \\ 0 & \boldsymbol{0}_{1\times L} \end{bmatrix}, \quad (14)$$

respectively. The d^{th} dimensional static feature components, $\mu_{m,t,d}^{(y|X)}$ and $\Sigma_{m,d}^{(y|X)}$, of the mean vector $\boldsymbol{\mu}_{m,t}^{(Y|X)}$ and the covariance matrix $\boldsymbol{\Sigma}_m^{(Y|X)}$ are used to predict the state covariance matrix and the state vector as shown in Eqs. (9) and (10). Their dynamic feature components, $\mu_{m,t,d}^{(\Delta y|X)}$ and $\Sigma_{m,d}^{(\Delta y|X)}$, are used to optimize the Kalman gain in Eq. (13) and update the state covariance matrix and the state vector as shown in (11) and (12). The first component of $\hat{\boldsymbol{y}}_d^{(t)}$ is used as the d^{th} component of the converted static feature vector at frame $t-L$, $\hat{y}_{t-L,d}$.

Note that the length of frame delay is $L+C$, where C is the number of succeeding frames in Eq. (1). This recursive update does not cause significant degradation in quality of the converted speech even if setting L to a small value, *e.g.*, 3 [4].

3. Implementation of Real-Time Voice Conversion Processes

3.1. Postfiltering with Global Variance

The global variance (GV) vector $\boldsymbol{v}^{(\boldsymbol{y})} = \left[v_1^{(\boldsymbol{y})}, \cdots, v_{D^{(y)}}^{(\boldsymbol{y})}\right]^\top$ is calculated from a time sequence vector of the target static feature vectors $\boldsymbol{y}$ utterance by utterance as follows:

$$v_d^{(\boldsymbol{y})} \quad = \quad \frac{1}{T}\sum_{t=1}^{T}\left(y_{t,d} - \frac{1}{T}\sum_{\tau=1}^{T}y_{\tau,d}\right)^2, \quad (15)$$

where $y_{t,d}$ is the d^{th} dimensional component of the target static feature vector $\boldsymbol{y}_t$ at frame t. Muffled sounds in the converted speech is significantly reduced by determining the converted static feature vectors using an additional penalty term on the GV in Eq. (5) so that their GV is close to the GV calculated from the natural target speech parameters. However, because this determination process is performed with an iterative batch-type update using the gradient, it is not suitable for real-time voice conversion.

As a conversion method considering the GV without the iterative update, we propose the postfiltering based on the GV. As in the conventional method, the mean vector of the GV of the target speech parameters, $\boldsymbol{\mu}^{(v)} = \left[\mu_1^{(v)}, \cdots, \mu_{D^{(y)}}^{(v)}\right]^\top$, is calculated in advance. Additionally, the source feature vectors in training data is converted to the target speech parameters using the trained GMM and the mean vector of their GV, $\hat{\boldsymbol{\mu}}^{(v)} = \left[\hat{\mu}_1^{(v)}, \cdots, \hat{\mu}_{D^{(y)}}^{(v)}\right]^\top$, is also calculated in advance. Moreover, a bias value of the converted speech parameters over an utterance is calculated utterance by utterance and its mean value over all utterances, $\langle\hat{y}_d\rangle$, is also calculated. Using these statistics, the d^{th} dimensional component of the converted static feature vector is enhanced frame by frame as follows:

$$\hat{y}_{t,d}^{(\text{GV})} \quad = \quad \mu_d^{(v)^{\frac{1}{2}}}\hat{\mu}_d^{(v)^{-\frac{1}{2}}}\left(\hat{y}_{t,d} - \langle\hat{y}_d\rangle\right) + \langle\hat{y}_d\rangle. \quad (16)$$

3.2. Computationally Efficient Conversion Algorithm

3.2.1. Rapid source feature extraction

To extract high-quality speech parameters, the state-of-the-art analysis methods, such as STRAIGHT analysis [7], are effective but their computational cost is usually expensive. In speech parameter extraction of the target voice, these analysis methods should be used since quality of the target speech parameters directly affects quality of the converted speech. Moreover, the computationally expensive analysis does not need to be performed in conversion. On the other hand, in speech parameter extraction of the source voice, its computational cost directly affects the conversion time.

To significantly reduce the computational cost while keeping the converted voice quality high, we propose the use of a lower-quality speech parameter extracted with simple FFT analysis as the source feature and the conversion from such a lower-quality speech parameter of the source voice into the high-quality speech parameter of the target voice using the GMM trained with the joint feature vectors based on those source and target speech parameters.

3.2.2. Diagonalization of covariance matrices

In some VC applications, such as alaryngeal speech enhancement [1] or body-conducted speech enhancement [8], the use of full covariance matrices is essential since different types of

speech parameters are used for the source and target features. It makes the mixture component selection in Eq. (6) more computationally expensive. To significantly reduce its computational cost while keeping accuracy in the mixture component selection high enough, a diagonalization method of the covariance matrices is proposed inspired by the semi-tied covariance [9].

In this paper, we implement constrained maximum likelihood linear regression (CMLLR) [10] for the diagonalization. The joint *p.d.f.* is written as

$$P\left(\boldsymbol{X}_t, \boldsymbol{Y}_t | \boldsymbol{\lambda}^{(X,Y)}, \boldsymbol{A}, \boldsymbol{b}\right) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(\boldsymbol{X}_t; \hat{\boldsymbol{\mu}}_m^{(X)}, \hat{\boldsymbol{\Sigma}}_m^{(XX)}\right) \mathcal{N}\left(\boldsymbol{Y}_t; \boldsymbol{\mu}_{m,t}^{(Y|X)}, \boldsymbol{\Sigma}_m^{(Y|X)}\right), \quad (17)$$

where the mean vectors and the covariance matrices of only the *p.d.f.* of the source feature vector are approximately modeled as

$$\hat{\boldsymbol{\mu}}_m^{(X)} = \boldsymbol{A}^{-1}\boldsymbol{\mu}_m^{(X')} - \boldsymbol{A}^{-1}\boldsymbol{b}, \quad \hat{\boldsymbol{\Sigma}}_m^{(XX)} = \boldsymbol{A}^{-1}\boldsymbol{\Lambda}_m^{(X'X')}\boldsymbol{A}^{-\top}, \quad (18)$$

respectively. The original mixture-dependent full covariance matrix $\hat{\boldsymbol{\Sigma}}_m^{(XX)}$ is represented with the mixture-dependent diagonal covariance matrix $\boldsymbol{\Lambda}_m^{(X'X')}$ and the global full transformation matrix $\boldsymbol{A}$. Both the global CMLLR transform $\{\boldsymbol{A}, \boldsymbol{b}\}$ and the mixture-dependent parameters $\left\{\boldsymbol{\Lambda}_m^{(X'X')}, \boldsymbol{\mu}_m^{(X')}\right\}$ are optimized in the sense of maximum likelihood with the training data set in the same manner as adaptive training.

In conversion, the global transform is applied to not the model parameters but the source feature vector as follows:

$$\boldsymbol{X}'_t = \boldsymbol{A}\boldsymbol{X}_t + \boldsymbol{b}. \quad (19)$$

If the transformation matrix for the feature extraction in Eq. (1) is also full, the global CMLLR transform is applied to them in advance as follows:

$$\boldsymbol{E}' = \boldsymbol{A}\boldsymbol{E}, \quad \boldsymbol{f}' = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{b}, \quad (20)$$

and therefore, the computational cost in the conversion does not increase. Using the transformed source feature vector $\boldsymbol{X}'_t$, the mixture component selection process in Eq. (6) is written as

$$\hat{m}_t = \arg\max_m \alpha_m \sqrt{|\boldsymbol{A}|^2} \mathcal{N}\left(\boldsymbol{X}'_t; \hat{\boldsymbol{\mu}}_m^{(X')}, \hat{\boldsymbol{\Lambda}}_m^{(X'X')}\right). \quad (21)$$

Thanks to the diagonal covariance matrix $\hat{\boldsymbol{\Lambda}}_m^{(X'X')}$, the computational cost significantly decreases compared with the use of the full covariance matrix $\boldsymbol{\Sigma}_m^{(XX)}$.

3.3. Implementation of conversion process

Figure 1 shows an example of a real-time VC process by setting analysis window length to 25 ms, frame shift to 5 ms, the parameter C in the source feature extraction to 2, the parameter L in the low-delay conversion to 2, and the minimum value of converted F_0 to 70 Hz. In the feature extraction, 25 ms delay (half window length 15 ms and two preceding frames 10 ms) is needed to extract the source feature vector at frame t. In the low-delay conversion, the converted speech parameters at frame $t-2$ is determined, and therefore, 10 ms delay for two frames is needed. In waveform synthesis, a one-pitch mixed excitation signal is generated using a converted F_0 value and converted aperiodic components capturing frequency-dependent noise strength if a synthesized pitch mark stands at frame $t-2$, and then overlap-add is performed. Due to anti-causality of the excitation signal, the one pitch excitation signal generated at frame $t-2$ possibly affects the excitation signal at three preceding frames (15 ms) if the minimum F_0 value is set to 70 Hz (14.3 ms). Finally, the generated excitation signal at frame $t-5$, which is no longer affected by the next one-pitch mixed excitation signal, is filtered with the converted spectral parameter at the corresponding frame to generate a converted waveform signal. These processes are performed frame by frame. Totally 50 ms maximum delay exists in this example.

The maximum delay changes depending on the VC application. In the conversion from body-conducted unvoiced speech to a whispered voice [8], 15 ms delay in waveform synthesis is no longer necessary since white noise is used as the excitation signal. Even in the most complicated conversion, such as alaryngeal speech or body-conducted unvoiced speech to a natural voice [1, 8], the maximum delay caused by a typical setting ($C = 4$, $L = 3$) is 65 ms. We have confirmed that this conversion process in 16 kHz sampling runs in real time on a laptop PC (Intel Core 2 Duo P8400, 2.26 GHz).

4. Experimental Evaluations

We conducted experimental evaluations to demonstrate the effectiveness of the proposed GV-based postfiltering and diagonalization methods in the VC application of body-conducted speech enhancement [8].

4.1. Experimental Conditions

We simultaneously recorded body-conducted natural voices and natural voices uttered by four Japanese speakers (two males and two females) using non-audible murmur microphone and headset microphone. Each speaker uttered about 50 phoneme balanced sentences for training and about 105 newspaper article sentences for evaluation. The sampling frequency was 8 kHz.

The 0^{th} through 16^{th} mel-cepstral coefficients were used as a spectral feature. PCA from 9 frames around a current frame ($C = 4$) was used to extract 34-dimensional segment feature at each frame. The conversion from the segment feature of the body-conducted natural voice into the mel-cepstrum of the natural voice was performed. In synthesis, STRAIGHT mixed excitation and MLSA filter were used.

In the evaluation of the GV-based postfilter, an opinion test on speech quality was conducted. Six listeners evaluated quality of the converted speech by the low-delay conversion with/without the GV postfilter and the conventional batch-type conversion considering the GV. STRAIGHT analysis [7] was used in both source and target feature extraction. The number of mixture components was set to 64.

Moreover, the computationally efficient conversion methods were evaluated with mel-cepstral distortion used as an evaluation metric. Simple FFT analysis was used in the computationally efficient source feature extraction. To clarify the effect of the proposed diagonalization, we compared the following conditions; the use of full covariance matrices of 64 mixture components (Full), the use of only diagonal components of those matrices (Only diag), the use of diagonal components but the number of mixture components increased up to 250 (Diag), and the use of the proposed diagonalization of the 64 mixture components (CMLLR+AT).

4.2. Effect of GV-based postfiltering

Figure 2 shows mean opinion score (MOS) as a result of the opinion test. Compared with the batch-type conversion considering the GV ('Batch-type w/ GV'), the low-delay conversion setting the frame delay to 5 without the GV-based postfil-

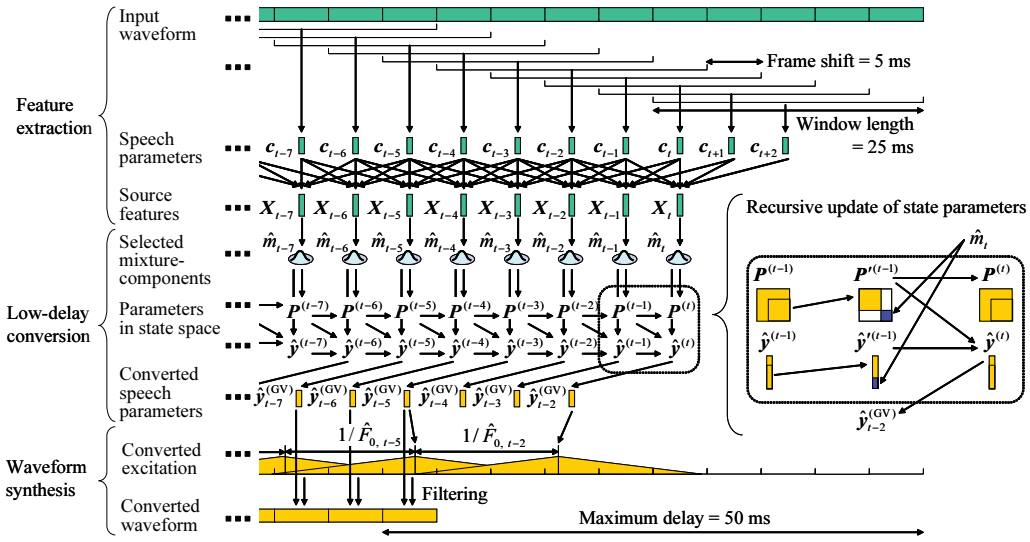

Figure 1: *Frame-by-frame processing in real-time voice conversion ($C = 2$, $L = 2$).*

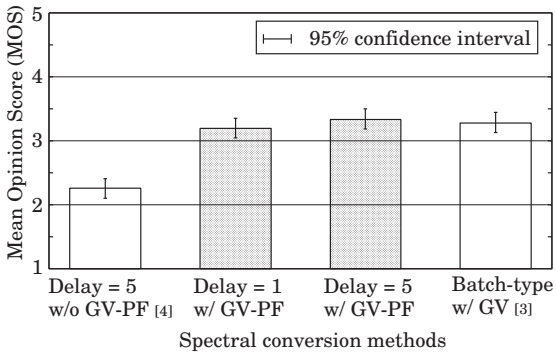

Figure 2: *Result of opinion test on speech quality.*

ter ('Delay = 5 w/o GV-PF') causes significant degradation in the converted speech quality. This degradation is not observed when using the GV-based postfiltering ('Delay = 5 w/ GV-PF'). Moreover, even if setting the frame delay to 1 ('Delay = 1 w/ GV-PF'), the converted speech quality is still comparable to the batch-type conversion.

4.3. Effect of Computationally Efficient Conversion

Table 1 shows the mel-cepstral distortion in each conversion setting. No degradation is observed by using FFT analysis instead of STRAIGHT analysis in the source feature extraction. If simply using only diagonal components of the full covariance matrices, significantly large degradation is caused. Its degradation is slightly reduced by increasing the number of mixture components but conversion accuracy is much worse than that by the full covariance matrices. We can see that the proposed diagonalization significantly reduces this degradation. We have found that the computational time in the proposed method is almost four times as fast as that in the conventional method.

5. Conclusions

This paper has presented an implementation of computationally efficient real-time voice conversion processing. Some experimental results have demonstrated that the proposed implementation yields good performance in both converted speech quality and computational complexity. We plan to further implement these techniques for digital signal processor (DSP).

Table 1: *Mel-cepstral distortion (MelCD).*

Analysis	Covariance	MelCD [dB]
STRAIGHT	Full (64 mix)	3.52
FFT	Full (64 mix)	3.52
FFT	Only diag (64 mix)	3.97
FFT	Diag (250 mix)	3.86
FFT	CMLLR+AT (64 mix)	3.59

Acknowledgment: This work was supported in part by MEXT Grant-in-Aid for Young Scientists (A).

6. References

[1] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano. Esophageal speech enhancement based on statistical voice conversion with Gaussian mixture models. *IEICE Trans. Inf. & Syst.*, Vol. E93-D, No. 9, pp. 2472–2482, 2010.

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

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

[4] T. Muramatsu, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano. Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *Proc. INTERSPEECH*, pp. 1076–1079, Brisbane, Australia, Sep. 2008.

[5] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. *Proc. of ICASSP*, pp. 660–663, Detroit, USA, May 1995.

[6] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi. Vector quantization of speech spectral parameters using statistics of dynamic features. *IEICE Trans. Information and Systems*, Vol. E84-D, No. 10, pp. 1427–1434, 2001.

[7] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F_0 extraction: possible role of a repetitive structure in sounds. *Speech Communication*, Vol. 27, No. 3–4, pp. 187–207, 1999.

[8] T. Toda, K. Nakamura, H. Sekimoto, and K. Shikano. Voice conversion for various types of body transmitted speech. *Proc. ICASSP*, pp. 3601–3604, Taipei, Taiwan, Apr. 2009.

[9] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Trans. Speech and Audio Processing*, Vol. 7, No. 3, pp. 272–281, 1999.

[10] 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.