Low-Delay Voice Conversion based on Maximum Likelihood Estimation of Spectral Parameter Trajectory

Takashi Muramatsu†, Yamato Ohtani†, Tomoki Toda†, Hiroshi Saruwatari†, and Kiyohiro Shikano†
†Graduate School of Information Science, NARA Institute of Science and Technology
8916-5, Takayama-cho, Ikoma-shi, Nara 630-0192 JAPAN
E-mail: †{takashi-mu, yamato-o, tomoki, sawatari, shikano}@is.naist.jp

Abstract

As typical voice conversion methods, two spectral conversion processes have been proposed: 1) the frame-based conversion that converts spectral parameters frame by frame and 2) the trajectory-based conversion that converts all spectral parameters over an utterance simultaneously. The former process is capable of real-time conversion but it sometimes causes inappropriate spectral movements. On the other hand, the latter process provides the converted spectral parameters exhibiting proper dynamic characteristics but a batch process is inevitable. To achieve the real-time conversion process considering spectral dynamic characteristics, we propose a time-recursive conversion algorithm based on maximum likelihood estimation of spectral parameter trajectory. Experimental results show that the proposed method achieves the low-delay conversion process, e.g., only one frame delay, while keeping the conversion performance comparably high to that of the conventional trajectory-based conversion.

Index Terms: speech synthesis, voice conversion, Gaussian mixture model, maximum likelihood estimation, time-recursive algorithm.

1. INTRODUCTION

Voice conversion (VC) is a technique for converting the source speech feature to the target one while keeping linguistic information. There are many VC applications for speech communication such as bandwidth expansion for telephone speech [1] and body-transmitted speech enhancement [2]. It is no doubt that a real-time conversion process is essential in these applications.

As a typical statistical approach to VC, the conversion method based on the Gaussian mixture model (GMM) has been proposed. In this method, the joint probability density of source and target features is modeled by a GMM, which is trained beforehand using a parallel data set consisting of several sentence pairs uttered by source and target speakers. The trained GMM allows the conversion from any sample of the source features into that of the target features.

So far there have been proposed two main GMM-based conversion methods: 1) the frame-based conversion method that converts spectral parameters frame by frame based on the minimum mean square error (MMSE) [3] and 2) the trajectory-based conversion method that converts all spectral parameters over an utterance simultaneously based on the maximum likelihood estimation (MLE) [4]. The former method is capable of the real-time (on-the-fly) conversion process because the source features at individual frames are converted independently of each other. However, this method sometimes causes spectral movements with inappropriate dynamic characteristics due to ignoring the feature correlation between frames. Consequently, the converted speech quality is often deteriorated. On the other hand, the latter method provides the converted spectral parameter sequence exhibiting proper dynamic characteristics by considering dynamic features of the converted parameters. This method causes significant quality improvements of the converted speech. However, it does not allow the real-time conversion process because the source features over an utterance need to be converted simultaneously for considering the feature correlation between frames.

In order to achieve the real-time conversion process considering dynamic characteristics of the converted parameters, we propose a time-recursive conversion algorithm based on MLE of spectral parameter trajectory. This method is inspired by the parameter generation algorithm for HMM-based speech synthesis [5] and the vector quantization algorithm for speech coding [6]. Experimental results demonstrate that the proposed method achieves the low-delay conversion process while keeping the conversion performance comparably high to that of the conventional trajectory-based conversion.

This paper is organized as follows. In Section 2, we describe the conventional VC methods. In Section 3, we describe the proposed low-delay VC method. In Section 4, we provide the experimental results. Finally, conclusions are given in Section 5.

2. Conventional GMM-Based Spectral Conversion Methods

2.1. Training of GMM

D-dimensional source and target feature vectors at frame $t$ are denoted by $x_t$ and $y_t$, respectively. The joint probability density of the source and target feature vectors is modeled as follows:

$$P(z_t | \lambda^{(s)}) = \sum_{m=1}^{M} w_m N(z_t; \mu_m^{(s)}, \Sigma_m^{(s)}),$$

where $z_t$ is a joint vector $[x_t, y_t]^{\top}$. The superscript $\top$ indicates transposition. The mixture component index is $m$. The total number of mixture components is $M$. The weight of the $m$-th mixture component is $w_m$. $N(\cdot; \mu, \Sigma)$ denotes the normal distribution with mean vector $\mu$ and covariance matrix $\Sigma$.

A parameter set of GMM $\lambda^{(s)}$ consists of weights, mean vectors and covariance matrices for individual mixture components. The mean vector $\mu_m^{(s)}$ and the covariance matrix $\Sigma_m^{(s)}$ of the $m$-th mixture component are written as

$$\mu_m^{(s)} = \left[ \begin{array}{c} \mu_m^{(x)} \\ \mu_m^{(y)} \end{array} \right], \quad \Sigma_m^{(s)} = \left[ \begin{array}{cc} \Sigma_{m}^{(xx)} & \Sigma_{m}^{(xy)} \\ \Sigma_{m}^{(yx)} & \Sigma_{m}^{(yy)} \end{array} \right].$$

Copyright © 2008 ISCA
Accepted after peer review of full paper

1076
September 22 – 26, Brisbane Australia
where $\mu_m^{(x)}$ and $\mu_m^{(y)}$ are the mean vectors of the $m$-th mixture component for the source and the target, respectively. The matrices $\Sigma_m^{(x)}$ and $\Sigma_m^{(y)}$ are the covariance matrices of the $m$-th mixture component for the source and target, respectively. The matrices $\Sigma_{m,t}^{(xy)}$ and $\Sigma_{m,t}^{(yx)}$ are the cross-covariance matrices of the $m$-th mixture component for the source and target, respectively. These parameters are estimated with the EM algorithm using the joint vectors extracted from the training data [7].

### 2.2. Frame-Based Conversion

The conditional probability density of $y_t$, given $x_t$, is represented as follows:

$$P(y_t|x_t, \lambda^{(x)}) = \sum_{m=1}^{M} P(m|x_t, \lambda^{(x)}) P(y_t|x_t, m, \lambda^{(x)}),$$

where

$$P(m|x_t, \lambda^{(x)}) = \frac{w_m N(x_t; \mu_m^{(x)}, \Sigma_m^{(x)})}{\sum_{m=1}^{M} w_m N(x_t; \mu_m^{(x)}, \Sigma_m^{(x)})},$$

$$P(y_t|x_t, m, \lambda^{(x)}) = N(y_t; E_{m,t}^{(y)}, D_{m}^{(y)}).$$

The mean vector $E_{m,t}^{(y)}$, and the covariance matrix $D_{m}^{(y)}$ of the $m$-th conditional probability distribution are written as

$$E_{m,t}^{(y)} = \mu_m + \sum_{m=1}^{M} \Sigma_{m,t}^{(xy)} \Sigma_{m}^{(x)^{-1}} (x_t - \mu_m^{(x)}),$$

$$D_{m}^{(y)} = \Sigma_m^{(y)} - \Sigma_m^{(yx)} \Sigma_m^{(x)^{-1}} \Sigma_m^{(x)}.$$

The MMSE-based conversion [3] is performed as follows:

$$\hat{y}_t = \sum_{m=1}^{M} P(m|x_t, \lambda^{(x)}) E_{m,t}^{(y)},$$

where $\hat{y}_t$ is the converted target feature vector. This method converts spectral parameters frame by frame.

### 2.3. Trajectory-Based Conversion

The target feature vector is defined as a 2D-dimensional vector $Y_t$ consisting of $D$-dimensional static and dynamic features at frame $t$. A relationship between the time sequences $Y = [Y_1^D, Y_2^D, \ldots, Y_T^D]^T$ and $y = [y_1^D, y_2^D, \ldots, y_T^D]^T$ is represented as $Y = Wy$ where

$$W = \begin{bmatrix} w_1^T, w_2^T, \ldots, w_T^T \end{bmatrix}^T,$$

and

$$w_t = [0_{D \times (t-2)D}, \bar{w}, 0_{D \times (T-t)D}],$$

$$\bar{w} = \begin{bmatrix} 0_{D \times D} & I_{D \times D} & 0_{D \times D} \\ -gI_{D \times D} & gI_{D \times D} & 0_{D \times D} \end{bmatrix}.$$ 

As the source feature vector $X_t$, the feature vector including both static and dynamic features [4] or the concatenated feature vector from multiple frames [8] is employed. This time sequence is written as $X = [X_1^T, X_2^T, \ldots, X_T^T]^T$. Using joint vectors $Z_t = [X_t^T, Y_t^T]^T$, the GMM parameter set $\lambda^{(2)}$ of the joint probability density $P(Z_t|\lambda^{(2)})$ is trained beforehand by adopting the conventional training framework [7].

In the conversion process, the converted feature vector sequence $\hat{y}$ is determined by maximizing the likelihood function $P(Y|X, \lambda^{(2)})$ as follows:

$$\hat{y} = \arg \max_{y} P(Y|X, \lambda^{(2)})$$

$$\simeq \arg \max_{y} P(\tilde{m}|X, \lambda^{(2)}) P(Y|X, \tilde{m}, \lambda^{(2)}).$$

### Figure 1: Schematic image of the trajectory-based conversion process.

where the suboptimal mixture component sequence $\tilde{m}$ is determined by maximizing the posterior probability $P(\tilde{m}|X, \lambda^{(2)})$. The ML estimate of $\hat{y}$ is given by solving the following linear equation

$$R\hat{y} = r,$$

where

$$R = P^{-1} = W^T D_m^{(y)_1} W,$$

$$r = W^T D_m^{(y)_1} E_m^{(y)},$$

and

$$E_m^{(y)} = \begin{bmatrix} E_m^{(y)_1}, \ldots, E_m^{(y)_t}, \ldots, E_m^{(y)_T} \end{bmatrix}^T,$$

$$D_m^{(y)_1} = \text{diag} \begin{bmatrix} D_m^{(y)_1}, \ldots, D_m^{(y)_t}, \ldots, D_m^{(y)_T} \end{bmatrix}.$$  

### 3. Proposed Low-Delay Conversion Method

We propose the time-recursive conversion algorithm by modifying the trajectory-based conversion in a manner similar to that described in [5] [6]. The following matrix $\overline{W}^{(t-1)}$ is obtained by replacing the elements of $W$, $\bar{w}$, with $\overline{w}$ from time $t$ to $T$.

$$\overline{W}^{(t-1)} = \begin{bmatrix} w_1^T, \ldots, w_{t-1}^T, w_t^T, w_{t+1}^T, \ldots, w_T^T \end{bmatrix}^T,$$

where

$$\overline{w}_t = \begin{bmatrix} 0_{D \times (t-2)D}, \overline{w}, 0_{D \times (T-t)D} \end{bmatrix},$$

$$\overline{w} = \begin{bmatrix} 0_{D \times D} & I_{D \times D} & 0_{D \times D} \\ -gI_{D \times D} & gI_{D \times D} & 0_{D \times D} \end{bmatrix}. $$

In this case, the set of equations corresponding to Eq. (13) can be written as

$$\overline{R}^{(t-1)} \overline{y}^{(t-1)} = \lambda^{(t-1)}.$$  

where

$$\overline{R}^{(t-1)} = \overline{R}^{(t-1)^T} D_m^{(y)_1} \overline{W}^{(t-1)},$$

$$\lambda^{(t-1)} = \overline{W}^{(t-1)^T} D_m^{(y)_1} E_m^{(y)},$$

$$\overline{y}^{(t-1)} = \begin{bmatrix} \overline{y}_1^{(t-1)^T}, \ldots, \overline{y}_t^{(t-1)^T}, \ldots, \overline{y}_T^{(t-1)^T} \end{bmatrix}^T.$$
Table 1: Summary of time-recursive algorithm

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q = J J^T + P_L$</td>
<td>(A.1)</td>
</tr>
<tr>
<td>$\psi = J y_L + y_L$</td>
<td>(A.2)</td>
</tr>
<tr>
<td>$\pi = Q v_L$</td>
<td>(A.3)</td>
</tr>
<tr>
<td>$\nu = v_L \pi$</td>
<td>(A.4)</td>
</tr>
<tr>
<td>$k = \pi (I_{2D \times 2D} + D_{m}^{(Y)})^{-1} \nu^{-1}$</td>
<td>(A.5)</td>
</tr>
<tr>
<td>$y_L^{(i)} = \psi + k D_{m}^{(Y)}^{-1} (E_{m,i-1} - v_L \psi)$</td>
<td>(A.6)</td>
</tr>
<tr>
<td>$P_L^{(i)} = Q - k D_{m}^{(Y)}^{-1} \pi^T$</td>
<td>(A.7)</td>
</tr>
</tbody>
</table>

The matrix $\overline{W}$ is defined so that $\overline{R}^{(0)}$ is a $D$-by-$D$ block diagonal matrix. When $\overline{w}_i$ is replaced with $\overline{w}_i$, the following equation is obtained

$$\overline{R}^{(i)} y^{(i)} = \overline{p}^{(i)} ,$$

where

$$\overline{W}^{(i)} = [ \overline{w}_1^T, \ldots, \overline{w}_{i-1}^T, \overline{w}_i^T, \overline{w}_{i+1}^T, \ldots, \overline{w}_T^T ]^T .$$

Using the above equations, the following equations are derived:

$$\overline{R}^{(t)} = \overline{R}^{(t-1)} + \nu^T \nu^{-1} \nu^T ,$$

$$\overline{p}^{(t)} = \overline{p}^{(t-1)} + \nu^T \nu^{-1} \nu^T ,$$

$$v_t = \nu^T \nu^{-1} \nu^T .$$

Using the matrix inversion lemma as in the derivation of the RLS algorithm [9], $\overline{y}^{(i)}$ and $\overline{P}^{(i)}$ are calculated recursively from $\overline{y}^{(i-1)}$ and $\overline{P}^{(i-1)}$ by the solution of Eq. (13), $y$. All elements of $\overline{y}^{(i)}$ are updated by replacing $\overline{w}_i$ with $\overline{w}_i$. Note that $\overline{y}^{(0)}$ is equal to the static parts of $E_{m}^{(Y)}$. By replacing $\overline{w}_i$ from time 1 to $T$, the final updated vector $\overline{y}^{(T)}$ is obtained, which corresponds to the solution of Eq. (13), $y$.

Table 1 shows the time-recursive algorithm derived by further introducing the sliding window concept of the RLS algorithm to the above recursive algorithm. In this algorithm, the sizes of the recursively updated parameters, $y_L^{(i)}$ and $P_L^{(i)}$, are limited to $LD$-by-1 and $LD$-by-$LD$, respectively. Therefore, the converted vectors obtained by this algorithm are suboptimal. Equations (A.1) and (A.2) are the key parts of this algorithm. Old elements are thrown out by using the matrix $J_a$ written as

$$J_a = \begin{bmatrix} 0_{(L-1) \times D} & I_{(L-1) \times D} & 0_{0 \times (L-1) D} \\ 0_{D \times D} & 0_{(L-1) \times D} \\ 0_{D \times (L-1) D} & \end{bmatrix} ,$$

and new components are added by using the $LD$-by-$LD$ matrix $P_L^{(i)}$ and the $LD$-by-1 vector $\overline{y}_i^{(T)}$, which are respectively written as

$$P_L^{(i)} = \text{diag} \left[ 0_{D \times D}, \ldots, 0_{D \times D}, \overline{R}_i^{-1} \right] ,$$

$$\overline{y}_i^{(T)} = [ 0, \ldots, 0, \overline{R}_i^{-1} \overline{r}_i ]^T .$$

The new components, the $D \times D$ matrix $\overline{R}_i$ and the $D \times 1$ vector $\overline{r}_i$ are written as

$$\overline{R}_i = J_a \overline{w}_i^T D_{m}^{(Y)} \overline{W}_L J_a^T ,$$

$$\overline{r}_i = J_a \overline{w}_i^T D_{m}^{(Y)} \overline{E}_{m,i}^{(Y)} ,$$

where the $2D \times LD$ matrices $\overline{w}_i, w_L$ and $v_L$, and the $LD \times LD$ matrix $J_a$ are respectively written as follows:

$$\overline{w}_i = [ 0_{2D \times D}, \ldots, 0_{2D \times D} ] ,$$

$$w_L = [ 0_{2D \times D}, \ldots, 0_{2D \times D} ] ,$$

$$v_L = w_L - \overline{w}_L ,$$

$$J_a = [ 0_{D \times D}, \ldots, 0_{D \times D}, I_{D \times D} ] .$$


Figure 2: Schematic image of the low-delay conversion process with $D = 1$.

The value $L$ is the number of frames stored in this algorithm, which is determined by

$$L = \max(D, 1) + 1 ,$$

where $D$ is the length of frame delay allowed in the spectral conversion. The converted vector at time $t$, $\hat{y}_t$ is obtained as

$$\hat{y}_t = \overline{y}_t - \overline{d}_{t-1} ,$$

where $\overline{y}^{(i)}$ is a vector contained in $y_L^{(i)}$ written as

$$\overline{y}_L^{(i)} = \left[ y^{(i)}_L - (L-1), \ldots, y^{(i)}_L \right]^T .$$

Figure 2 shows the schematic image of this conversion process with $D = 1$. The statistics of conditional probabilities at the previous frames are effectively propagated to the next frame. Consequently, the converted feature vectors are determined frame by frame while considering the statistics of conditional probabilities at the current and succeeding $D$ frames as well as all previous frames. In the proposed algorithm, the statistic $E_{m,i}^{(Y)}$ in (A.6) of Table 1 continuously changes frame by frame because it is represented as the linear transformation of the source feature vectors unlike the algorithms described in [5] [6].

It is noted that this time-recursive algorithm is valid if the cross-covariance values of static and dynamic features in $D_{m}^{(Y)}$ are zeros. If using full elements of $D_{m}^{(Y)}$, it is necessary to derive the time-recursive algorithm based on a replacement of elements of $D_{m}^{(Y)}$ instead of $\overline{W}^{(i)}$ in a manner similar to that described in [5].

4. Experimental Evaluations

The performance of the proposed method is evaluated in the framework of body-transmitted VC [2]. There have been proposed voice conversion frameworks for improving the quality of several types of body-transmitted speech. In this paper, we conduct the conversion from body-transmitted ordinary speech (BTOS) into natural speech. This conversion system is very effective for achieving noise-robust speech communication.

4.1. Experimental Conditions

We simultaneously recorded BTOS and natural speech uttered by the same speaker using a NAM microphone [2], which is one of the body-conductive microphones, and an air-conductive microphone. Sampling frequency was set to 8 kHz.

The 0-th through 16-th mel-cepstral coefficients were used as a spectral feature. The frame shift was set to 5 ms. In order to compensate missing acoustic features of BTOS due to the body transmission effects as described in [2], we used the spectral segment vector as the source feature, which was constructed by
concatenating the mel-cepstra at a current ±4 frames and then reducing the concatenated vector dimension using PCA with a loss of no more than 20% of the information. STRAIGHT [10] analysis was employed for the target spectral extraction. On the other hand, simple FFT analysis was employed for the source spectral extraction for reducing the processing time. In our preliminary experiment, it was shown that the different spectral analysis method for the source speech doesn’t cause any significant differences of the conversion performance.

We used four speakers including two male and two female. Table 2 shows the number of training/evaluation sentences in each speaker. The conversion model was trained for each speaker separately. The number of mixture components was set to 64.

### 4.2. Objective Evaluations

We used the mel-cepstral distortion between the target and converted mel-cepstra as the objective evaluation measure.

Figure 3 shows the mel-cepstral distortions in the frame-based, trajectory-based and proposed methods, respectively. The proposed method outperforms the conventional frame-based method even in the no-delay condition because it effectively considers the dynamic features of the converted spectra. The distortion quickly decreases as the delay value increases, and it is almost equal to that in the trajectory-based method when setting the delay value to around five.

### 4.3. Subjective Evaluations

We conducted a preference test on speech quality. We evaluated converted speech by two conventional methods (the frame-based and the trajectory-based) and by the proposed methods when setting the delay value to 0, 1, and 5, respectively. A pair of the converted voices from two different methods was randomly presented to the five listeners, and then they were asked which voice sounded more natural.

Figure 4 shows the preference scores of individual methods with confidence intervals (95%). The converted speech quality of the proposed method with no delay is almost equal to that of the frame-based conversion. It is significantly improved by introducing some delay values, even only one frame (i.e., 5 ms). Consequently, the proposed method achieves a very low-delay frame-by-frame conversion process while keeping the converted speech quality comparably high to that of the trajectory-based conversion.

### 5. Conclusions

In this paper, we proposed a spectral conversion method for high-quality and low-delay VC. The experimental results show that the proposed method of spectral conversion by the time-recursive algorithm achieves high-quality and low-delay conversion. We plan to reduce the computational cost of the conversion process and the required memory size to develop a small run-time VC system for embedded applications.

### 6. Acknowledgments

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

### 7. References


---

**Table 2: Number of sequences for training/evaluation**

<table>
<thead>
<tr>
<th>speaker</th>
<th>male A</th>
<th>male B</th>
<th>female C</th>
<th>female D</th>
</tr>
</thead>
<tbody>
<tr>
<td>training</td>
<td>93</td>
<td>89</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>evaluation</td>
<td>46</td>
<td>45</td>
<td>47</td>
<td>47</td>
</tr>
</tbody>
</table>

**Figure 3: Mel-cepstral distortion in each spectral conversion method.**

**Figure 4: Result of preference tests of speech quality.**