Sliding Window-based Speech-to-Lips Conversion with Low Delay

Wei Han*†, Lijuan Wang†, Frank Soong†, Bo Yuan*
*Shanghai Jiao Tong University, Shanghai
E-mail: weihan_cs@sjtu.edu.cn,yuanbo@cs.sjtu.edu.cn
†Microsoft Research Asia, Beijing
E-mail: {lijuanw,frankkps}@microsoft.com

***Abstract*—The goal of a good speech-to-lips conversion system is to synthesize high quality, realistic lips movement which is time synchronized with the input speech. Previously, the maximum probability estimation of visual trajectory by Gaussian Mixture Model (GMM) has been successfully proposed and tested for speech-to-lips conversion. It works as a sentence level batch process that convert acoustic speech signals to visual lips movement trajectory. In this paper, we propose a moving window based, low delay speech-to-lips conversion method for real-time communication applications. The new approach is an approximation of the MLE-GMM conversion but can render lips movement on-the-fly with a low time latency. Experimental results on the LIPS2009 dataset shows that proposed real-time method can achieve a latency of less than 100ms while maintain comparable quality as the batch method.**

I. Introduction

Speech-driven lips conversion aims at synthesizing lip movements that are consistent and time synchronous with given human speech signals. It has a wide range of practical applications in multimedia communication as well as human-computer interactions. For example, an avatar-based video phone can be implemented by transferring only the speech signals and the corresponding talking head can be rendered at the receiving end with a very low data rate of "hidden" information. Personalized speech-driven avatar can also be rendered in video games or other augmented reality scenario.

Two main aspects that can affect the practical use of this technique: the intelligibility of the visual lips movements and the latency which is the delay when users see the output lips animation after their speech arrives the system. Low latency is a crucial requirement in certain applications of lip synthesis such as video conference.

Various approaches have been proposed for the synthesis of lips movement from speech. However, most of them focus on the quality of visual output but neglect the latency part, and few can at the same time achieve both high-quality lips animation and low latency.

Earlier work in the literature includes phoneme/viseme mapping and key-frame interpolation which work on a per-frame basis [1]. They are suitable for real-time applications, but the quality of the rendered lips movements is not as good as later methods in terms of intelligibility and smoothness [2]. Hidden Markov Model (HMM) and its variants have also been widely used for predict lips movement from speech. These methods [2] depend on a set of HMMs, using model sequence

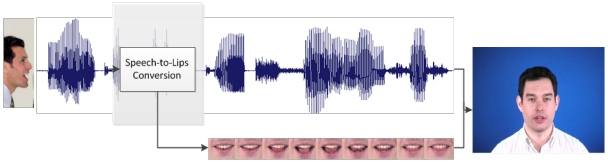

Fig. 1. An illustration of the speech-driven lips conversion system. The input is speech only and the output is audio-visual signal

as an intermediate stage which requires Viterbi decoding that brings segmentation and phone identification errors. An EM-like procedure named HMM inversion (HMMI) has also been proposed [3], which is too computationally intensive for low-cost real-time applications.

Another approach in speech-to-lips rendering is to use Gaussian Mixture Models (GMM). In this method, the joint probability density of audio-visual features is modeled by a GMM. Two different ways to perform speech-to-lips conversion with GMM are: 1) The frame-by-frame conversion subject to minimum mean square error [4]; 2) The maximum probability estimation of trajectory [5] [6], which use both static and dynamic feature statistics. While the former method works on each frame and is suitable for real-time applications, it may lead to converted trajectories with inaccurate dynamic characteristic without imposing inter-frame dynamic constraints in the conversion. On the other hand, the maximum probability method by incorporating dynamic statistics have been shown to significantly improve the quality of converted trajectory. However, it needs all acoustic feature frames in an utterance to be available and processed simultaneously, thus hinders its usage in real-time applications.

In this paper, we modify the batch conversion algorithm by introducing a sliding time window and performing the computation in a recursive sense. We evaluated the algorithm on LIPS2009 Visual Speech Synthesis Challenge Task [7]. The proposed algorithm shows the quality of rendered lips movements is comparable to the batch method in [6] in terms of the commonly accepted objective metrics.

The rest of the paper is organized as follows. In Section 2 we give an overview of the batch processing needed conversion method. In Section 3 we propose the sequential sliding window method for rendering for low delay real-time applications. In Section 4 we present experimental results. In Section 5 we draw our conclusions.

II. Speech-to-Lips Conversion

As illustrated in Fig. 1, a speech-to-lips conversion system can render realistic lips movements that are in sync with input human voice. Speech-to-lips conversion core module plays a central role in mapping acoustic signals to a video sequence. The statistical correlation between audio and visual space is exploited in such a module. In the literature, it is often named audio-visual mapping or conversion.

A. Audio-visual Modeling with GMM

We define the acoustic and visual feature vector sequences as $\mathbf{x} = [x_1^\top, \ldots, x_T^\top]^\top$ and $\mathbf{y} = [y_1^\top, \ldots, y_T^\top]^\top$, respectively. Audio-visual conversion defines a mapping function from $\mathbf{x}$ to $\mathbf{y}$. The joint probability density of x_t and y_t can be modeled statistically as a Gaussian mixture model (GMM)

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

where z_t is the augmented super-vector $[x_t^\top, y_t^\top]^\top$, m is the index of the m-th mixture component, and w_m is the weight of m-th mixture component. In the pdf, μ and Σ denotes the mean vector and covariance matrix of a Gaussian component distribution, which can be written explicitly in terms of x and y as:

$$\mu_m = \begin{bmatrix} \mu_m^{(x)} \\ \mu_m^{(y)} \end{bmatrix}, \Sigma_m = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix} \tag{2}$$

$\lambda = \{\mathbf{w}, \mu, \mathbf{\Sigma}\}$ denotes for all parameters for the joint audio-visual GMM. It can be estimated from the training data by maximizing the likelihood (ML), or other training criteria like minimum converted trajectory error (MCTE) proposed which has been shown to achieve better performance [6].

B. MLE-based Conversion

MLE-based conversion was first proposed for voice conversion [5], and was later adopted for speech-to-lips conversion [6]. In this method, the D-dimensional vector y_t is extended to $2D$-dimensional vector Y_t as $Y_t = [y_t^\top, \Delta y_t^\top]^\top$. The sequence $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_T]$ could be represented as a linear transform of the static (instantaneous observation) vectors $\mathbf{y}$, $\mathbf{Y} = W\mathbf{y}$, such that $\Delta y_t = \frac{1}{2}(y_{t+1} - y_{t-1})$. In a similar way, $\mathbf{x}$ is extended to $\mathbf{X}$.

The conditional probability density is formulated as,

$$\begin{aligned} P(\mathbf{Y}|\mathbf{X}, \lambda) &= \sum_{\text{all } \mathbf{m}} P(\mathbf{m}|\mathbf{X}, \lambda) P(\mathbf{Y}|\mathbf{X}, \mathbf{m}, \lambda) \\ &\approx \prod_{t=0}^{T} \sum_{m=1}^{M} P(m|X_t, \lambda) P(Y_t|X_t, m, \lambda) \end{aligned} \tag{3}$$

The conversion is performed by maximize the conditional probability to obtain the estimate vector $\hat{\mathbf{y}}$

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(W\mathbf{y}|\mathbf{X}, \lambda) \tag{4}$$

In practice, we make several approximations to reduce the complexity in solving Eq. 4. First, the summation in Eq. 3 is approximated by the Maximum A Posterior (MAP) mixture component, $\hat{m}_t$,

$$P(\mathbf{Y}|\mathbf{X}, \lambda) \approx \prod_{t=0}^{T} P(\hat{m}_t|X_t, \lambda) P(Y_t|X_t, \hat{m}_t, \lambda) \tag{5}$$

where $\hat{m}_t = \arg\max_m P(m|X_t, \lambda)$. With this approximation, Eq. 4 can then be solved in a closed form,

$$\hat{\mathbf{y}} = (W^\top {D_{\hat{m}}^{(Y)}}^{-1} W)^{-1} W^\top {D_{\hat{m}}^{(Y)}}^{-1} E_{\hat{m}}^{(Y)} \tag{6}$$

where

$$E_{\hat{m}}^{(Y)} = \left[E_{\hat{m}_1,1}^{(Y)}, \ldots; \ldots; \ldots, E_{\hat{m}_T,T}^{(Y)} \right] \tag{7}$$

$${D_{\hat{m}}^{(Y)}}^{-1} = \text{diag}\left[{D_{\hat{m}_1}^{(Y)}}^{-1}, \ldots; \ldots; \ldots, {D_{\hat{m}_T}^{(Y)}}^{-1} \right] \tag{8}$$

and

$$E_{\hat{m}_t,t}^{(Y)} = \mu_{\hat{m}_t}^{(Y)} + \Sigma_{\hat{m}_t}^{(YX)} {\Sigma_{m_t}^{(XX)}}^{-1} (X_t - \mu_{\hat{m}_t}^{(X)}) \tag{9}$$

$$D_{\hat{m}_t}^{(Y)} = \Sigma_{\hat{m}_t}^{(YY)} - \Sigma_{\hat{m}_t}^{(YX)} {\Sigma_{\hat{m}_t}^{(XX)}}^{-1} \Sigma_{\hat{m}_t}^{(XY)} \tag{10}$$

Second, to have a robust estimation of covariance matrix Σ, we assume that, the off-diagonal terms $\Sigma_m^{(XY)}$ and $\Sigma_m^{(YX)}$ are all null matrices, and $\Sigma_m^{(XX)}$ and $\Sigma_m^{(YY)}$ are diagonal. In other words, correlations between different dimensions in the joint space (X, Y) are ignored. Eventually, Eq. 9 and 10 are simplified to

$$E_{\hat{m}_t,t}^{(Y)} \approx \mu_{\hat{m}_t}^{(Y)} \quad D_{\hat{m}_t}^{(Y)} \approx \Sigma_{\hat{m}_t}^{(YY)} \tag{11}$$

Note that $E_{\hat{m}_t,t}^{(Y)}$ and $D_{\hat{m}_t}^{(Y)}$ require for all $\hat{m}_t$. Consequently, the conversion requires all acoustic frames $\mathbf{X}$ of an utterance to be available before solving the simultaneous equations to obtain the relation in Eq. 6. This is usually impossible in real-time.

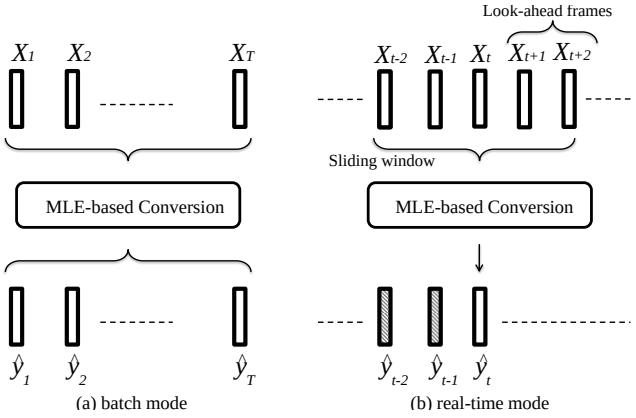

Fig. 2. *Batch and real-time (sliding window) speech-to-lips conversion*

III. Low-Delay Conversion

The *latency* denotes the time delay between an available acoustic frame and the rendered corresponding lips movement. A long latency can cause inconvenience in real-time communications between human and machine. Therefore, a low latency is highly desirable for many practical talking head applications.

As illustrated in Fig. 2, in a batch mode MLE-based conversion, all acoustic features in an utterance are converted simultaneously. The total system latency is the length of acoustic utterance plus computation time in all stages.

We want to achieve a low latency for real-time applications, but retain high quality of rendered lips movements. For this purpose, we propose a sliding window-based real-time speech-to-lips conversion method.

A. Sliding Window-based Real-time Conversion

For real-time processing of sequence $[X_1, X_2, \ldots, X_T]$, $\hat{y}_t$ needs to be rendered as soon as X_t is available. While batch conversion waits until all frames in X_T become available to obtain $\hat{y}_t$, we can approximate it using only $[X_1, X_2, \ldots, X_t]$ that are available at time t, solve the batch conversion problem in these frames. Then y_t can be obtained sequentially with time.

Furthermore, frames X_i in the long past often have few influence on X_j than the recent frames. A sliding window approach that considers only the most recent L frames is appropriate to reduce computation load and hopefully without deteriorating the output quality appreciably.

In the sliding window approach, we define

$$E^{(Y)}_{\hat{m},t,L} = \left[E_{\hat{m}_{t-L+1},t-L+1}{}^{(Y)}, \ldots; \ldots; \ldots, E_{\hat{m}_t,t}{}^{(Y)}\right] \quad (12)$$

$$D^{(Y)}_{\hat{m},t,L} = diag\left[D_{\hat{m}_{t-L+1}}^{(Y)}, \ldots; \ldots; \ldots, D_{\hat{m}_t}^{(Y)}\right] \quad (13)$$

as the statistics in a sliding window ranges from frame index $t-L+1$ to t. Batch conversion in section 2 can be performed in the window as:

$$\begin{aligned}\hat{\mathbf{y}}_{t,L} &= \arg\max P(W_L \mathbf{y}_{t,L} | \mathbf{X}_{t,L}) \\ &= (W_L^\top D^{(Y)}_{\hat{m},t,L}{}^{-1} W_L)^{-1} W_L^\top \\ &\quad D^{(Y)}_{\hat{m},t,L}{}^{-1} E^{(Y)}_{\hat{m},t,L} \end{aligned} \quad (14)$$

and have the estimation of $\hat{y}_t$:

$$\hat{y}_t = \hat{\mathbf{y}}_{t,L}(L) \quad (15)$$

In practice, we observed that if a few frames come after X_t can provide important dynamic constraints between $\hat{y}_t$ and $\hat{y}_{>t}$, thus improve the estimation of trajectory $\hat{\mathbf{y}}$. Therefore, we allow *look-ahead* frames, that is, to wait until X_{t+h} become available before estimate $\hat{y}_t$. Here h is the number of look-ahead frames.

$$\hat{y}_t = \hat{\mathbf{y}}_{t+h,L}(L-h) \quad (16)$$

As illustrated in Fig. 2, the methods work as follows. At the time $t+h$, L most recent frames are put into sliding window. Then batch conversion is performed on the sequence inside this window by Eq. 14, gives a local trajectory $\hat{\mathbf{y}}_{t+h,L}$. In this local trajectory we keep only the L-th item $\hat{\mathbf{y}}_{t+h,L}(L)$ as the estimation of $\hat{y}_t$ and discard the rest. In the next step, the window slides forward for one frame and solve for $\hat{y}_{t+1}$.

Initially at $t=0$, $E^{(Y)}_{\hat{m},t+h,L}$ and $D^{(Y)}_{\hat{m},t+h,L}{}^{-1}$ are set to be all zero. When new frame become available, we update the two matrix in a recursive manner similar to [8],

$$E^{(Y)}_{\hat{m},t+h+1,L} = JE^{(Y)}_{\hat{m},t+h,L} J^\top + \Delta E_{t+h+1} \quad (17)$$

$$D^{(Y)}_{\hat{m},t+h+1,L}{}^{-1} = JD^{(Y)}_{\hat{m},t+h,L}{}^{-1} J^\top + \Delta D_{t+h+1} \quad (18)$$

where elements no longer in the window are thrown out by J,

$$J = \begin{bmatrix} 0_{LD\times D} & I_{LD\times LD} \\ 0_{D\times D} & 0_{D\times LD} \end{bmatrix} \quad (19)$$

and new components is added by,

$$\Delta E_{t+h+1} = \left[0, \ldots; \ldots; \ldots, E^{(Y)}_{\hat{m}_{t+h+1},t+h+1}\right] \quad (20)$$

$$\Delta D_{t+h+1} = diag\left[0, \ldots; \ldots; \ldots, D^{(Y)}_{\hat{m}_{t+h+1}}{}^{-1}\right] \quad (21)$$

Eq. 14 can be solved by Cholesky decomposition as derived in [9]. It operates on the order of $\mathcal{O}(LM)$ if the covariance matrix is diagonal. M is the dimensionality of Y.

In this method, the synthesis of $\hat{y}_t$ still makes use of the inter-frame constraints brought about by dynamic features, but limited in the window. In other words, the proposed approach use local solution in Eq. 14 to approximate the utterance-wide solution in Eq. 6. In our observation, with reasonable window length and look-ahead frames (we use $L = 200$ and $h = 10$), the synthesized trajectories will be very closed to there counterpart from batch conversion.

Fig. 2 indicates the latency of this method: the length of look-ahead frames plus computation time for solving Eq. 14 in sliding window. Although increasing the latency, look-ahead frames can significantly improve the quality of conversion, as mentioned in [8] and observed in our experiments. Therefore we tolerate a small number of look-ahead frames, which can be treated as a trade-off between the latency of system and the fidelity of output trajectory.

IV. Evaluations

A. Experimental Setup

We employ the LIPS 2008/2009 Visual Speech Synthesis Challenge data [7] to evaluate the proposed conversion method. The dataset has 278 video files with the corresponding audio tracks, each is a separately recorded English sentence spoken neutrally by a female native speaker. The video frame rate is 50 frames per second. For each image in a video sequence, Principle Component Analysis (PCA) projection is performed on the automatically detected and aligned mouth image, resulting in a reduced, 60-dimensional visual parameter vector. Mel-frequency Cepstral Coefficient (MFCC) vectors of the acoustic speech signals are extracted with a 20ms time window shifted every 5ms. The visual parameter vectors are interpolated up to the same frame rate as the audio speech MFCCs.

We compare the performance of proposed low-delay speech-to-lips conversion method with the batch conversion. In addition, we evaluate both the performance and the latency of real-time conversion with different length of sliding windows and different number of look-ahead frames. The trade-off between high rendering quality and low time latency in this system is therefore investigated.

B. Objective Evaluation

The performance of conversion is evaluated objectively by two metrics, in two, open and close, tests. In the "close" test we use all available data for both training and conversion while in the "open" test, we use a leave-20-out cross validation experiment and the errors averaged over all the folds to calibrate the performance.

	MSE		ACC	
	open	close	open	close
Batch	7.69e5	5.27e5	0.406	0.598
Real-time	9.11e5	7.21e5	0.400	0.547

TABLE I
OBJECTIVE EVALUATIONS, LOOK-AHEAD=10 FRAMES

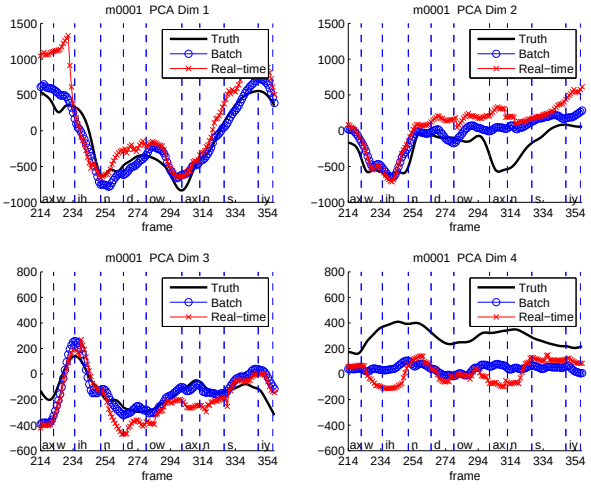

Fig. 3. Trajectory of Top PCA dimensions

We use Mean Square Error (MSE) and Average Correlation Coefficient (ACC) as objective performance measures. They are defined as,

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}||\hat{y}_t - y_t|| \tag{22}$$

$$\text{ACC} = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{(y_{t,d}-\mu_{y_d})(\hat{y}_{t,d}-\mu_{\hat{y}_d})}{\sigma_{y_d}\sigma_{\hat{y}_d}} \tag{23}$$

In Table I, we observe the performance degradation which results from the approximation used in the real-time system. Fig. 3 shows the converted visual trajectory between the batch and real-time sequential conversion. Compared with batch conversion, the trajectory synthesized in a sliding window has a larger distance away from ground truth, resulting in a larger MSE. Nevertheless, it still retains a smooth trajectory and more importantly, a shape that is very similar to the ground truth. In other words, the proposed method can reproduce proper dynamic characteristics in its output trajectory. This can also be demonstrated by the ACC measure, in which the difference between batch and real-time conversion is rather small.

C. Rendering Performance versus Latency

As discussed in Section 3, the look-ahead parameter serves as a trade-off between high rendering performance and low time latency. To investigate the effects of look-ahead frames quantitatively, we perform the objective evaluation with different look-aheads. As shown in Fig. 4, the MSE is large without look-ahead but decreasing rapidly as look-ahead number increases.

In practical, we use 10 look-ahead frames as a compromise of rendering performance and latency. Since the frame shift is 5ms, latency induced by look-ahead will be 50ms. Other possible causes of latency includes:

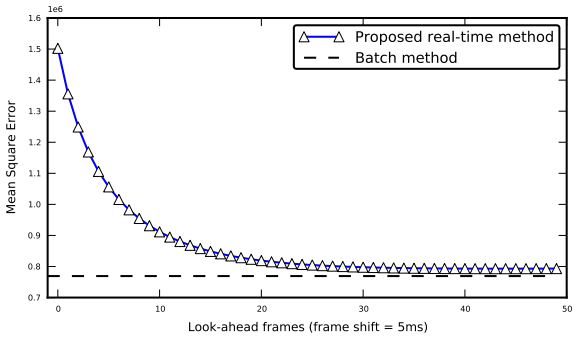

Fig. 4. Performance under different number of look-ahead frames

- Acoustic feature and its delta parameter extraction also needs a short time window, the length is set at 20ms in our experiments.
- Time for computation. This is minor since the computation of the proposed algorithm is linear on L and M.

As a final performance assessment, we ran the complete speech-to-lips system on a PC with live input of audio speech signals. The actual latency turns out to be less than 100 ms, and the audio-visual delay is almost negligible to human viewers.

V. CONCLUSION

In this paper, we modify the batch GMM-based, speech-to-lips conversion method for low latency, real-time applications, with a sliding window covering the most recent frames and a look-ahead. Experimental results on the LIPS 2009 dataset demonstrate the effectiveness of the proposed method. It can approach the performance of the previous batch method in both the quality of synthesized trajectory and low latency. Look-ahead frames can be set as a control parameter that can compromise synthesized trajectory fidelity and the resultant latency.

REFERENCES

[1] C. Bregler, M. Covell, and M. Slaney, "Video Rewrite: driving visual speech with audio," in *Proc. ACM SIGGRAPH '97*, 1997, pp. 353–360.

[2] S. Fu, R. Gutierrez-Osuna, A. Esposito, P. K. Kakumanu, and O. N. Garcia, "Audio/visual mapping with cross-modal hidden markov models," *IEEE Trans. on Multimedia*, vol. 7, no. 2, pp. 243–252, 2005.

[3] K. Choi, Y. Luo, and J. Hwang, "Hidden Markov model inversion for audio-to-visual conversion in an MPEG-4 facial animation system," *The Journal of VLSI Signal Processing*, vol. 29, no. 1, pp. 51–61, 2001.

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

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

[6] X. Zhuang, L. Wang, F. K. Soong, and M. Hasegawa-Johnson, "A minimum converted trajectory error (MCTE) approach to high quality speech-to-lips conversion," in *INTERSPEECH*, 2010, pp. 1736–1739.

[7] B.-J. Theobald, S. Fagel, G. Bailly, and F. Elisei, "LIPS2008: visual speech synthesis challenge," in *INTERSPEECH*, 2008, pp. 2310–2313.

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

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *ICASSP*, 2000, pp. 1315–1318.