Fast Locally Linear Embedding Algorithm for Exemplar-based Voice Conversion

Yu-Huai Peng*, Chin-Cheng Hsu†, Yi-Chiao Wu†, Hsin-Te Hwang†, Yi-Wen Liu*, Yu Tsao‡, and Hsin-Min Wang†
* National Tsinghua University, Hsinchu, Taiwan
† Institute of Information Science, Academia Sinica, Taipei, Taiwan
‡ Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
E-mail: roland19930601@gmail.com, ywliu@ee.nthu.edu.tw, yu.tsao@citi.sinica.edu.tw, whm@iis.sinica.edu.tw

***Abstract*—The locally linear embedding (LLE) algorithm has been proven to have high output quality and applicability for voice conversion (VC) tasks. However, the major shortcoming of the LLE-based VC approach is the time complexity (especially in the matrix inversion process) during the conversion phase. In this paper, we propose a fast version of the LLE algorithm that significantly reduces the complexity. In the proposed method, each locally linear patch on the data manifold is described by a pre-computed cluster of exemplars, and thus the major part of on-line computation can be carried out beforehand in the off-line phase. Experimental results demonstrate that the VC performance of the proposed fast LLE algorithm is comparable to that of the original LLE algorithm and that a real-time VC system becomes possible because of the highly reduced time complexity.**

I. Introduction

Voice conversion (VC) is a technique that transforms one type of speech to another, without changing the linguistic content. A typical application is speaker VC, which modifies a source speaker's speech to sound as if it is spoken by a target speaker. Generally speaking, speaker VC involves spectral and prosodic conversions. In this study, we focus on spectral conversion, whereas a simple linear transformation of F0 is applied for prosodic conversion in our VC system. So far, a variety of approaches have been proposed to tackle VC, such as the Gaussian mixture model (GMM) [1], [2], frequency warping [3], [4], neural networks (NN) [5], [6], and exemplar-base methods [7], [8], [9].

Recently, we have proposed an exemplar-based VC approach [9] based on the locally linear embedding (LLE) algorithm [10], which is a manifold learning algorithm. The LLE algorithm is readily applicable to VC when a parallel speech corpus is available because the source-target relation is intuitively linked. Assuming that the local structures of the source speech space and that of the target speech space are similar, we can use LLE to find the relationship between an input source spectral feature vector and its neighboring exemplars and apply this relationship to the corresponding target exemplars to obtain the desired target spectral feature vector. Note that the LLE-based VC approach can be applied on-the-fly once the parallel dictionary is available; all of the remaining computation is executed during the on-line (conversion) stage.

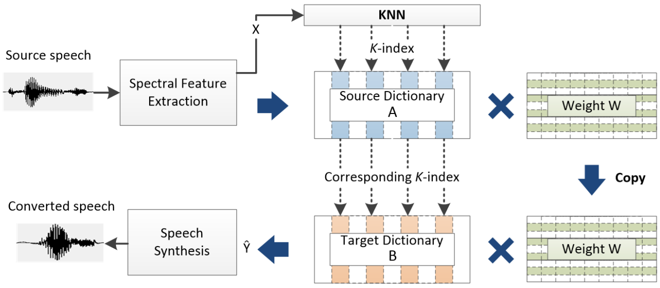

Fig. 1: The LLE-based VC framework.

The advantage of this non-parametric approach is that it does not require training. However, the shortcoming is that the LLE-based VC approach is computationally demanding during the conversion phase, leading to difficulty for real-time applications. To address this problem, we propose an accelerated version of the LLE algorithm called *fast-LLE* that utilizes exemplars while enjoying faster conversion.

The remainder of this paper is organized as follows. We briefly summarize LLE-based VC in Sec. II, and introduce the proposed fast-LLE algorithm and fast-LLE-based VC in Sec. III. The experimental results are presented in Sec. IV. Finally, we differentiate the fast-LLE algorithm with related works in Sec. V, and conclude the paper in Sec. VI.

II. LLE-based VC

The simplest version of conventional exemplar-based VC from a parallel corpus seeks to convert the voice characteristics on a frame-by-frame basis. Suppose that we have N pairs of speech frames; every pair is made of one frame $\mathbf{a}_i$ from the source speaker and one corresponding frame $\mathbf{b}_i$ that has same or similar linguistic content from the target speaker. Let $A = \{\mathbf{a}_i\}_{i=1}^N$ denote the source dictionary, $B = \{\mathbf{b}_i\}_{i=1}^N$ denote the target dictionary, and $X = \{\mathbf{x}_t\}_{t=1}^T$ denote the input sequence (an utterance of T frames) from the source speaker. Let A_t denotes a *local subset* of A with respect to a given (external) vector $\mathbf{x}_t$ and B_t denotes the corresponding subset of B. Note that both A_t and B_t are made of column vectors and the dimensions are both D (the feature dimension) by K (the number of nearest neighbors). The goal is to convert $\mathbf{x}_t$ into $\mathbf{y}_t$ that has the target speaker's voice characteristics.

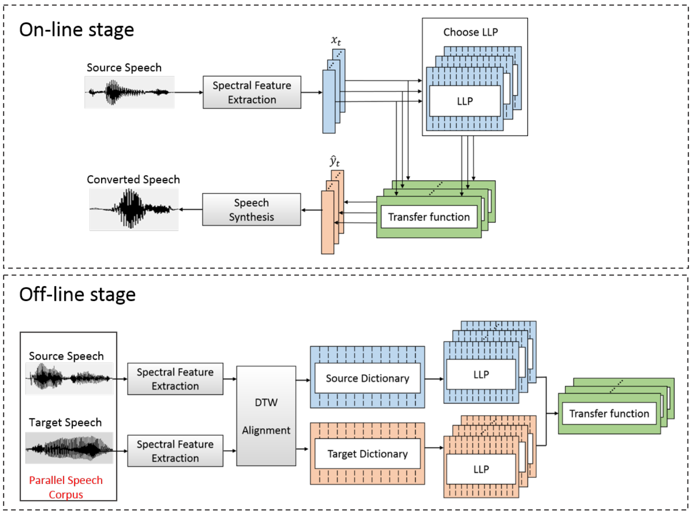

Fig. 2: Flow charts of the on-line and off-line stages of the fast-LLE VC system. The transfer function is the same as (8).

The LLE-based VC approach [9] integrates the LLE algorithm [10] with a conventional exemplar-based VC approach; it assumes that the geometry of A_t is locally linear and resembles that of B_t. Exploiting data parallelism and geometry resemblance, we can approach conversion via reconstruction:

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \end{bmatrix} \approx \begin{bmatrix} A_t \\ B_t \end{bmatrix} \mathbf{w}_t. \tag{1}$$

To be more specific, we first find the weight vector that minimizes the reconstruction error between $\mathbf{x}_t$ and $A_t\mathbf{w}_t$; then we apply the same $\mathbf{w}_t$ to B_t to obtain the converted frame $\mathbf{y}_t$.

Fig. 1 depicts the system architecture of LLE-based VC. We construct a pair of parallel dictionaries A and B from a parallel speech corpus in the off-line stage. During the on-line (conversion) stage, given a speech utterance of the source speaker, the LLE algorithm is employed to perform frame-by-frame conversion in the following three steps:

1) Identifying the locally linear patch (LLP) A_t from the source dictionary A by finding a set of K nearest neighbors (using the k-nearest neighbors (k-NN) algorithm) with respect to the input frame $\mathbf{x}_t$.
2) Characterizing the local geometry by solving a weight vector $\mathbf{w}_t$ that minimizes the reconstruction error $\|\mathbf{x}_t - A_t\mathbf{w}_t\|^2$.
3) Converting the input frame to the output frame by applying the same $\mathbf{w}_t$ to the corresponding target LLP B_t, i.e., $\hat{\mathbf{y}}_t = B_t\mathbf{w}_t$.

III. Fast-LLE VC

In LLE-based VC, the three steps in the on-line stage are conducted sequentially, i.e., we have to determine A_t, the LLP with respect to an input frame $\mathbf{x}_t$, solve the weight $\mathbf{w}_t$, and finally convert the frame. However, solving the reconstruction weight is highly time consuming, preventing us from a real-time application.

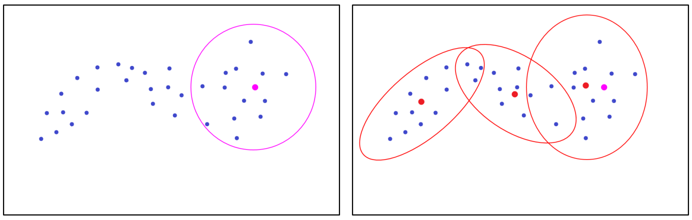

Fig. 3: LLPs in fast-LLE. Blue dots are exemplars, red dots are the cluster centroids, and magenta dots are the inputs. Left: the LLP identified by LLE. Right: the LLP identified by fast-LLE. In fast-LLE, the LLP with respect to the input is defined by the set of exemplars whose cluster centroid is closest to the input.

In order to accelerate the conversion, we propose *fast-LLE* – a simplification that allows most of the computation to be done off-line. The keystone of *fast-LLE* is to approximate the data manifold by a fixed number of pre-defined LLPs. With pre-defined LLPs, we can compute necessary matrix inversions during the off-line stage, thereby reducing time complexity by orders of magnitude in the on-line conversion stage.

A. *The Modification Made in fast-LLE*

As shown in Fig. 2, the initial step of the off-line procedures is identical to that of LLE-based VC, which is to build a source-target paired dictionary. The fast-LLE algorithm diverges from the LLE algorithm afterwards with the following steps:

1) The data manifold is scaffolded by M anchors which can be cluster centroids or simply samples from the dictionary. We used the cluster centroids determined by k-means in this study.
2) For each anchor, an accompanying LLP is built by retrieving the K nearest neighbors with respect to it (measured in the Euclidean distance). These exemplars then become the bases of the LLP (Fig. 3). Note that LLPs can have overlapping exemplars, i.e., one exemplar can belong to multiple LLPs.
3) Each anchor represents the accompanying LLP; the distance between an input $\mathbf{x}_t$ and an anchor is regarded as the proxy of the distance between the input and an LLP.

By doing so, we can replace the computation required for a k-NN search from the whole dictionary of cardinality N with a 1-NN search from a candidate set of cardinality M during the on-line stage.

Note that even in the extreme case where $M = N$, the fast-LLE algorithm still differs from the LLE algorithm in that fast-LLE retrieves K nearest neighbors with respect to an anchor instead of the input itself.

B. *The Statistics in fast-LLE (off-line stage)*

Consider the solution of LLE-based VC. The loss function is defined as

$$\mathcal{L} = \|\mathbf{x}_t - A_t\mathbf{w}_t\|^2 + \lambda(\mathbf{1}^{\mathrm{T}}\mathbf{w}_t - 1), \tag{2}$$

TABLE I: Comparison of the computational complexity for the conversion of a frame.

	LLE	fast-LLE
Identifying LLP	k-NN $O(DN + KN)$	1-NN $O(DM)$
Weight estimation	$O(DK^2 + K^3)$	None
Conversion	$O(DK)$	$O(D^2)$
Complexity	$O(DN + KN + DK^2 + K^3)$	$O(D^2 + DM)$

D: frame dimension (72 in our case).
K: number of nearest neighboring exemplars in LLE, which is 1024 in our case.
M: number of LLPs generated in the off-line stage, which is 128 in our case.
N: number of exemplars in the (parallel) dictionary, typically several tens of thousands. It is about 11000 in our case.

where λ is the Lagrange multiplier of the constraint and $\mathbf{1}$ is an all-one vector of length K. We can derive an analytic solution by setting the derivatives $\frac{\partial \mathcal{L}}{\partial \mathbf{w}_t}$ to zero:

$$\mathbf{w}_t = (A_t^{\mathrm{T}} A_t)^{-1}(A_t^{\mathrm{T}} \mathbf{x}_t + \lambda \mathbf{1}), \tag{3}$$

$$\lambda = \frac{1 - \mathbf{1}^{\mathrm{T}}(A_t^{\mathrm{T}} A_t)^{-1} A_t^{\mathrm{T}} \mathbf{x}_t}{\mathbf{1}^{\mathrm{T}}(A_t^{\mathrm{T}} A_t)^{-1}\mathbf{1}}. \tag{4}$$

Although we still have to identify to which LLP the input $\mathbf{x}_t$ belongs to, the corresponding LLP A_t is now a fixed set of exemplars, as opposed to a dynamically determined set with combinatorially many possibilities in LLE.

The computation that can be carried out during the off-line stage is described as follows. First, we can reformulate (3) as

$$\mathbf{w}_t = D_t \mathbf{x}_t + \mathbf{e}_t, \tag{5}$$

where

$$D_t = (A_t^{\mathrm{T}} A_t)^{-1} A_t^{\mathrm{T}} - \frac{(A_t^{\mathrm{T}} A_t)^{-1}\mathbf{1}\mathbf{1}^{\mathrm{T}}(A_t^{\mathrm{T}} A_t)^{-1} A_t^{\mathrm{T}}}{\mathbf{1}^{\mathrm{T}}(A_t^{\mathrm{T}} A_t)^{-1}\mathbf{1}}, \tag{6}$$

and

$$\mathbf{e}_t = \frac{(A_t^{\mathrm{T}} A_t)^{-1}\mathbf{1}}{\mathbf{1}^{\mathrm{T}}(A_t^{\mathrm{T}} A_t)^{-1}\mathbf{1}}. \tag{7}$$

The statistics D_t and $\mathbf{e}_t$ can be computed off-line because they do not depend on the input $\mathbf{x}_t$.

In addition, from (1) and (5), we have

$$\hat{\mathbf{y}}_t = B_t \mathbf{w}_t = B_t D_t \mathbf{x}_t + B_t \mathbf{e}_t := D_t^{(y)} \mathbf{x}_t + \mathbf{e}_t^{(y)}. \tag{8}$$

We can further save a matrix multiplication by pre-computing $D_t^{(y)} = B_t D_t$ and $\mathbf{e}_t^{(y)} = B_t \mathbf{e}_t$, and take (8) as the transfer function in Fig. 2. We compare the complexity of the on-line stages in LLE-based VC and fast-LLE-based VC in Table I.

C. Conversion

During the on-line stage, we first identify to which LLP an input $\mathbf{x}_t$ belongs, by finding the LLP whose centroid is closest to $\mathbf{x}_t$. Then we fetch the pre-computed statistics $D_t^{(y)}$ and $\mathbf{e}_t^{(y)}$ and use the transfer function (8) to convert $\mathbf{x}_t$ into $\hat{\mathbf{y}}_t$.

IV. Experiments

A. The Speech Corpus

Our experiments were conducted on the Sinica COSPRO speech corpus [11]. The corpus contained 9 datasets. The intonation-balanced dataset (i.e., COSPRO 03) consisting of Mandarin parallel speech utterances of 3 females and 2 males was used in the experiments. There were 20 pairs of conversions: 8 intra-gender and 12 inter-gender. For each conversion pair, 10 utterance pairs were randomly selected as the training set, 40 as the development set, and 43 as the test set. Speech signals were recorded in 16 kHz 16-bit wav format. Silence segments at the start and end of each utterance in the training set were discarded based on the segmentation information in the corpus.

B. Feature Extraction and Waveform Generation

We used the STRAIGHT vocoder [12] for feature extraction and waveform generation. During feature extraction, the speech signals were parametrized into the smoothed spectral envelopes (SEs), aperiodicity components (APs), and F0 contours. The frame shift was 5 milliseconds. The FFT length was set to 1024; thus, the AP and SE vectors for each frame were 513-dimensional. We further extracted the 24-order mel-cepstral coefficients (MCCs) from the SEs of each frame. The static, delta, and delta-delta features were used. Accordingly, the final MCC feature vector of a frame was 72-dimensional. LLE (or fast-LLE)-based conversion was applied to the MCC feature vectors. The linear mean-variance transformation was used to convert the log F0. The 0-th MCC and the APs were kept unmodified. The converted MCCs were reverted back to the SEs. Finally, the converted SEs, converted F0, and source APs were passed to the STRAIGHT vocoder for waveform synthesis.

C. The VC Systems with Post-processing

We compared two VC systems:

- LLE (baseline): Our previously proposed LLE-based VC system [9].
- fast-LLE (proposed): The proposed VC system that is based on the accelerated LLE algorithm.

For both systems, the number of nearest neighbors for the LLE algorithm was set to 1024 following our previous work in [9].

To further improve the quality of the converted speech, for both systems, we applied two post-processing procedures to the converted features, including the global variance (GV) post-filtering method [13] and the maximum likelihood parameter generation (MLPG) algorithm [14], [15].

D. The Effect of the Number of Clusters

We adopted Mel-cepstral distortion (MCD) as the the objective evaluation measure in the experiments. The MCD between a target frame and a converted frame is defined as follows:

$$\mathrm{MCD}\ [\mathrm{dB}] = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}\left(mc_d^{(y)} - \hat{mc}_d^{(y)}\right)^2}, \tag{9}$$

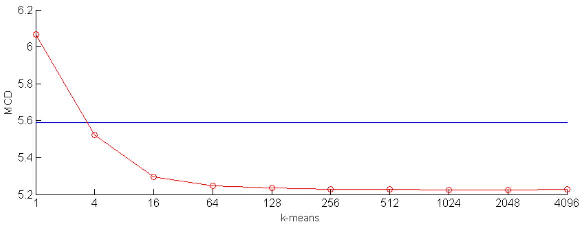

Fig. 4: Mean MCD versus the number of clusters (i.e., pre-generated LLPs). The blue line is the result of the LLE baseline system and the red curve is the result of the fast-LLE system.

TABLE II: Comparison of computational time.

Method	MCD	Speed (seconds per frame)
Baseline (LLE)	5.5907	6.09e-2
fast-LLE	5.2355	9.63e-5

where $mc_d^{(y)}$ and $\hat{mc}_d^{(y)}$ is the d-th element of the target and converted MCC vector, respectively [16]. A lower MCD indicates less spectral distortion. The MCD of an utterance pair was obtained by averaging over the MCDs of all the frame pairs in the utterance. We reported the mean MCD of all the test utterance pairs.

Fig. 4 shows the mean MCD versus the number of clusters (i.e., pre-generated LLPs), M. As M increased, the mean MCD was lowered; however, when M reached 128, the mean MCD started to saturate. Therefore, we set M to 128 in the following experiments. Surprisingly, the mean MCD of the fast-LLE system was lower than that of the LLE baseline system. This result indicated that using k-NN to identify an LLP might be suboptimal in LLE. However, the difference in mean MCD did not reflect significance in subjective evaluations, as will be discussed later.

E. Subjective Evaluation

For the subjective evaluation, we randomly selected two conversion pairs from each category (including f-f, m-m, m-f, and f-m; m: male, f: female), resulting in eight conversion pairs. For each conversion pair, eight sentences were randomly selected from the test set, thereby resulting in 64 (8x8) test sentences. Ten Chinese-native listeners were recruited to conduct the speech quality and speaker similarity tests.

1) Speech Quality: We conducted a mean opinion score (MOS) test to evaluate the quality of the converted speech. Specifically, each pair of converted speeches by systems **LLE** and **fast-LLE** were presented in a random order to the listeners. The listeners were asked to judge which sample sounded more natural and to grade them from 1 (bad) to 5 (good) points. Fig. 5 shows the overall average results of the preference test. The performance of the fast-LLE system is indistinguishable from that of the LLE baseline system, indicating that the fast-LLE algorithm is a good simplification of the LLE algorithm.

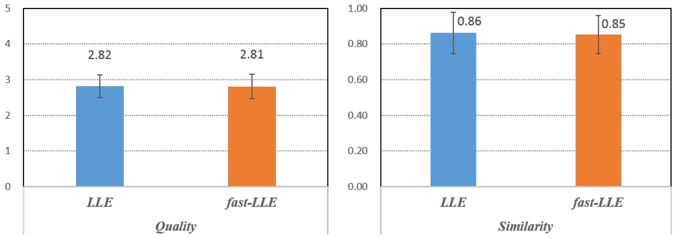

Fig. 5: Subjective test result. The error bars indicates confidence intervals.

2) Speaker Similarity: We conducted an ABX test for each system independently to evaluate the speaker similarity performance. In an ABX test, we presented 3 utterances to a listener: utterances A, B, and X. Utterances A and B were a pair of natural utterances from the source speaker and the target speaker with the same linguistic content in a shuffled order; utterance X was a converted speech with different linguistic content from utterances A and B so as to prevent listeners from exploiting the prosodic cues [16]. Listeners were asked to judge whether the speaker identity is the same as utterance A or utterance B. Note that we only reported the results of intra-gender conversion because all the inter-gender conversion pairs were identified correctly in our preliminary result. Similar results have also been reported in [8]. As shown in Fig. 5, the two methods achieved almost the same performance in terms of speaker similarity.

In summary, the performance of the fast-LLE system in terms of both voice quality and speaker similarity is indistinguishable from that of the LLE baseline system in the experiments. Since the fast-LLE algorithm has a computation time less than the frame shift in feature extraction (see Table II), it facilitates real-time VC applications.

V. Related Works

The fast-LLE VC system is very similar to the GMM-based VC system [1], [2] in the sense that both are mixture models. While GMM assigns soft labels to each data point upon computing statistics, fast-LLE imposes hard labels based on locality. In fast-LLE, a patch is regarded as locally linear and all the K members contribute equally to the statistics; in contrast, data points that do not belong to this LLP contribute none to the statistics. A final empirical result worth mentioning is that the LLE-based VC system has been proven to be effective even when the dimensionality of spectral features is high [9]. But the GMM-based VC system could only work on the low-dimensional spectral features.

Another kin to the fast-LLE algorithm is the exemplar-based non-negative matrix factorization (ENMF) method. The ENMF-based VC system [17] imposes an additional sparsity constraint to ensure better quality of synthetic speech. In addition, ENMF demands a large dictionary to have a satisfactory quality of synthetic speech at the cost of slow conversion. In contrast, we have demonstrated that it is beneficial to choose exemplars with a cluster-based scheme and that the computation can be drastically reduced without deteriorating the performance.

VI. Conclusions

We have proposed the fast-LLE algorithm as an accelerated alternative to the LLE algorithm and successfully applied it to realize a real-time VC system. The fast-LLE algorithm defined a number of locally linear patches that allowed us to shift most computation in the on-line conversion stage to the off-line stage, making real-time applications possible. The experimental results demonstrated that the performance of the fast-LLE VC system remained as good as that of the LLE VC system. For future extensions, we plan to explore the fast-LLE algorithm in more theoretical details and apply the fast-LLE algorithm to many-to-many VC and speech enhancement tasks.

References

[1] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion," *Proc. ICASSP*, pp. 4859–4863, 2015.

[2] H.-T. Hwang, Y. Tsao, H.-M. Wang, Y.-R. Wang, and S.-H. Chen, "Incorporating global variance in the training phase of GMM-based voice conversion," *Proc. APSIPA ASC*, pp. 1–6, 2013.

[3] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 18, no. 5, pp. 922–931, Jul. 2010.

[4] E. Godoy, O. Rosec, and T. Chonavel, "Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora," *IEEE Trans. Audio, Speech, Lang., Process*, vol. 20, no. 4, pp. 1313–1323, May 2012.

[5] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 22, no. 12, pp. 1859–1872, 2014.

[6] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Dictionary update for NMF-based voice conversion using an encoder-decoder network," *Proc. ISCSLP*, pp. 1–5, Oct. 2016.

[7] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," *Proc. SLT Workshop*, pp. 313 – 317, 2012.

[8] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 22, no. 10, pp. 1506–1521, Oct. 2014.

[9] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, "Locally linear embedding for exemplar-based spectral conversion," *Proc. INTERSPEECH*, pp. 1652–1656, 2016.

[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[11] C.-Y. Tseng, Y.-C. Cheng, and C.-H. Chang, "Sinica COSPRO and toolkit– corpora and platform of Mandarin Chinese fluent speech," *Proc. Oriental COCOSDA*, pp. 23–28, 2005.

[12] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, no. 3-4, pp. 187–207, 1999.

[13] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to implement global variance in statistical speech synthesis," *Proc. INTERSPEECH*, pp. 1436–1439, 2012.

[14] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," *Proc. ICASSP*, pp. 660–663, 1995.

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

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

[17] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, "Exemplar based voice conversion using non-negative spectrogram deconvolution," *Proc. 8th ISCA Speech Synth. Workshop (SSW8)*, pp. 201–206, 2013.