On the Use of I-vectors and Average Voice Model for Voice Conversion without Parallel Data

Jie Wu*, Zhizheng Wu† and Lei Xie*
* Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
E-mail: {jiewu, lxie}@nwpu-aslp.org
† The Centre for Speech Technology Research, University of Edinburgh, UK
E-mail:wuzhizheng@gmail.com

***Abstract*—Recently, deep and/or recurrent neural networks (DNNs/RNNs) have been employed for voice conversion, and have significantly improved the performance of converted speech. However, DNNs/RNNs generally require a large amount of parallel training data (e.g., hundreds of utterances) from source and target speakers. It is expensive to collect such a large amount of data, and impossible in some applications, such as cross-lingual conversion. To solve this problem, we propose to use average voice model and i-vectors for long short-term memory (LSTM) based voice conversion, which does not require parallel data from source and target speakers. The average voice model is trained using other speakers' data, and the i-vectors, a compact vector representing the identities of source and target speakers, are extracted independently. Subjective evaluation has confirmed the effectiveness of the proposed approach.**

Index Terms: voice conversion, nonparallel training, average voice model, i-vector, long short-term memory

I. Introduction

Voice conversion (VC) is a technique to modify the voice spoken by one speaker (source) so that it sounds like it is spoken by another speaker (target) while retaining its linguistic information [1] [2]. VC technology can be applied to various tasks, such as personalized text-to-speech (TTS) synthesis system [3], emotion conversion [4], speech enhancement [5], movie dubbing, and other entertainment applications.

Many statistical parametric approaches have been studied so far, mainly including linear and nonlinear feature mappings. Gaussian mixture models (GMMs) were proposed to implement weighted linear conversion functions [6]. Toda *et al.* [7] improved GMM-based method using dynamic features and the global variance (GV). Alternatively, dynamic kernel partial least squares (DKPLS) technique was proposed to model nonlinearity of the inherent time-dependency between speech features [8]. Non-parametric approaches have also been proposed such as exemplar-based non-negative matrix factorization (NMF) [9] [10], which directly used the target speech exemplars to synthesize the converted speech. However, most of the conventional methods, such as GMM, DKPLS and NMF, are based on "shallow" voice conversion architectures, in which the spectral feature of source speech are converted directly in the original feature space.

To capture the characteristics of speech more precisely, it might be more appropriate to have several hidden layers in the conversion architecture. Deep neural networks (DNNs), aiming to learn hierarchical feature mappings layer by layer, match this goal perfectly. As an early attempt, Desai *et al.* [11] have shown that neural network with multiple layers significantly outperforms GMM in both objective and subjective evaluations. Nakashika *et al.* [12] proposed a voice conversion method using deep belief nets (DBNs) in a high order eigenspace. Chen *et al.* [13] have also confirmed the superior performance of DNNs in the voice conversion task. Recently, in order to take advantage of the speech context information, Sun *et al.* [14] proposed a deep bidirectional long short-term memory (DBLSTM) architecture for voice conversion, which elegantly captures both frame-wised and long-range correlations between source and target features. Their work demonstrates that DBLSTM achieves superior performance than a feed-forward DNN. However, the above mentioned methods require a set of parallel data[1] from source and target speakers to train the mapping function. To achieve reasonable performance, the number of sentence pairs may expand to several hundreds [14]. It is expensive and even impossible to collect such parallel data in real applications, and the requirement of parallel data becomes a bottleneck to the practical use of voice conversion.

There have been some approaches that do not require parallel data between the source and target speaker for conversion. During model training, they usually *borrow* parallel data from speakers at hand (e.g., a corpus available with multiple speakers), to train a basic mapping function. Then the speech data from a new source-target speaker pair, which is unnecessarily paralleled, is used to adapt the base model. To this end, approaches can be approximately grouped into two categories: maximum a posterior (MAP) adaptation and eigenvoice conversion (EVC). In MAP adaptation [15] [16], target speech is used to adapt a source GMM for voice conversion without parallel data. Toda *et al.* [17] introduced eigenvoice conversion, originally proposed for speaker adaptation in speech recognition [18], into voice conversion, by adapting the conversion model using many pre-stored speakers' voices. In order to further improve the performance of converted speech with non-parallel data, Ohtani *et al.* [19] proposed an adaptive

[1] The source and target speakers read the same sentence, i.e., a parallel pair.

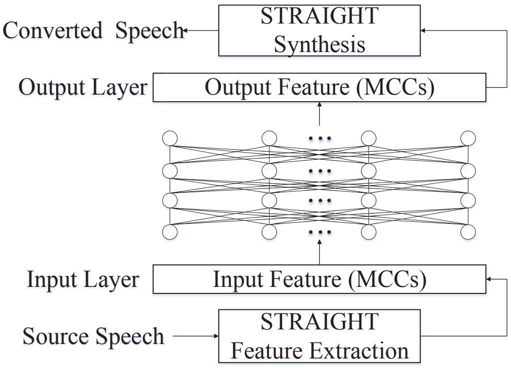


Fig. 1. The framework of DBLSTM based voice conversion approach.

training method for eigenvoice Gaussian mixture model (EV-GMM).

In this paper, following the success of deep bidirectional LSTM, we propose to use average voice model (AVM) and *i-vectors* for NN-based voice conversion without the source-target parallel data. Specifically, inspired by the previous work on GMM adaptation, we use a large amount of parallel utterances from other speakers as prior data to train a DBLSTM-based average voice model (AVM). But different in AVM training, we augment the source spectrum with source and target speaker identity vectors as network input, to learn a universal mapping to the spectrum of the target speaker. In this work, speaker identity is represented by i-vector[2], a low-dimensional speaker-specific vector extracted in a text-independent fashion. Without an adaptation stage, the AVM+i-vector model can be directly used for conversion: given the spectrum of a new source speaker and his/her i-vector, along with the i-vector of the new target speaker, the network directly generates the corresponding spectrum for the new target speaker. Subjective evaluation has confirmed the effectiveness of our proposed approach.

II. DBLSTM FOR VOICE CONVERSION

The flow diagram of a typical DBLSTM-based voice conversion system is shown in Fig. 1 [14]. In this study, we use STRAIGHT [21] to extract mel-cepstral coefficients (MCCs) for source and target speech, respectively. Usually, a voice conversion system is composed of a training and a conversion stage. During the training process, we use a frame alignment method, i.e., dynamic time warping (DTW), to get the parallel utterances between the source and target speakers. Then, the DBLSTM model is trained by the back-propagation through time (BPTT) algorithm. A nonlinear relationship between aligned spectral features (MCCs) of source and target speech is thus learned. This can be formulated as a nonlinear mapping function $F(\cdot)$ between the source spectral feature X and target spectral feature Y

$$\hat{Y} = F(X). \tag{1}$$

In the conversion stage, given the MCCs feature X of a new speech sequence from the source speaker, the corresponding converted MCCs $\hat{Y}$ for the target speaker are generated by the trained model in a frame-wised way. Finally, we use STRAIGHT as the vocoder to reconstruct the speech from the converted MCCs. Previous study has shown that this straightforward conversion approach achieves superior performance [14].

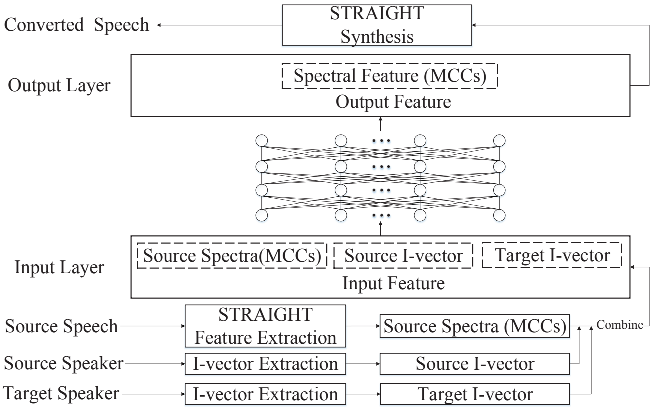


Fig. 2. The framework of average voice model with augmented i-vector input.

III. PROPOSED: AVM WITH AUGMENTED I-VECTORS

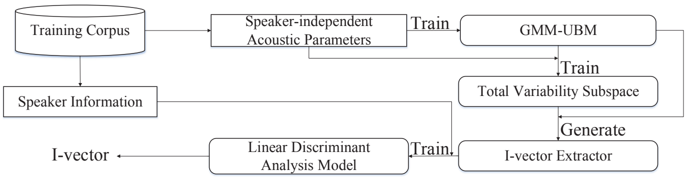
A. Basic Framework

The overall framework of our proposed approach is presented in Fig. 2, in which we augment i-vectors to the input feature in an average voice model. The network structure is actually the same with Fig. 1, i.e., using a DBLSTM as the mapping tool [22]. But the major differences lie in the network input and the way we train and use the model.

- **The network input**: In our proposed approach, i-vectors of the source and target speakers are augmented to the spectral feature of the source speech, which are used to capture the identity information of the source and target speakers.
- **The model training**: In the conventional DBLSTM approach, parallel utterances between source and target speakers are needed to train a specific conversion model. In contrast, in the proposed approach, we use parallel data of many other speakers to train an average voice model with i-vector inputs. We do not need the parallel data between the source and target speakers to train or re-train this model. In the conversion stage, given the spectrum of a new source speaker and his/her i-vector, along with the i-vector of the new target speaker, the average voice model directly generates the corresponding spectrum for the new target speaker.

In general, by combining the average voice model and the speaker identity information, we do not need the parallel utterances between the desired speakers in the model training. Thus, we can convert speech easily from a new source speaker

[2] I-vectors are widely used in speaker recognition [20].



Training Corpus
Speaker-independent Acoustic Parameters
Train
GMM-UBM
Train
Speaker Information
Total Variability Subspace
Generate
I-vector
Linear Discriminant Analysis Model
Train
I-vector Extractor

Fig. 3. The flow diagram of i-vector extraction.

to a new target speaker without their parallel utterances, which makes voice conversion more convenient in practical applications.

B. I-vector Extraction

In this work, we employ the framework introduced in [23] to generate i-vectors, which is shown in Fig. 3. I-vector is a low-dimensional vector representing speaker individuality and has been widely used in speaker recognition. In [20], a speaker super-vector (i.e., GMM super-vector) is projected onto a low-dimensional subspace based on factor analysis, resulting in an i-vector. Specifically, given an utterance from a single speaker, a speaker-dependent GMM super-vector is represented as

$$M = m + Tw, \tag{2}$$

where m is the speaker-independent super-vector[3] generated from a universal background model (UBM), a large GMM trained to represent the speaker-independent distribution of speech features. In other words, the UBM represents the inner structure of the whole acoustic space of a large number of speakers. M is the mean super-vector of the speaker-dependent GMM model adapted from the UBM, while T is the total variability matrix, which represents the speaker space on the background data. The speaker space is also called total variability subspace (TVS). Here, w is a random vector having a standard normal distribution $N(0, I)$, which is the so-called identity vector or *i-vector* for short, controlling the speaker individuality.

To make the i-vector more robust and compact, linear discriminant analysis (LDA) [24] is usually adopted. Given the i-vector generator mentioned above and the speaker identity labels, LDA aims to maximize the inter-class (inter-speaker) variance and minimize the intra-class (intra-speaker) variance, which has shown to be a key factor in the use of i-vector in speaker recognition [25]. Finally, a robust and low-dimensional i-vector is generated for an utterance from a certain speaker.

C. The Whole Process

According to the framework in Fig. 2, we summarize the whole process for using i-vectors in the average voice model. The training stage is as follows.

- **Data preparation**: Prepare a corpus with many speakers for AVM training, in which parallel utterances between source and target speakers are needed. But different source-target speaker pairs do not need speak the same set of sentences.
- **Feature extraction**: Use the vocoder (e.g., STRAIGHT) and the i-vector extractor to obtain the spectral features and the i-vectors of each source-target utterance pair in the corpus, respectively.
- **AVM training**: Following the framework in Fig. 2, train the DBLSTM model through BPTT using the DTW aligned parallel data. Note that the input of the network is the combination of spectral feature and the source and target i-vectors. Each speaker uses a fixed i-vector, which is averaged from the training utterances of the speaker.

The voice conversion process is quite straightforward. Given a new source-target speaker pair, which is not included in the AVM training, we firstly extract the spectral feature of source speech, source and target speaker i-vectors. The source and target i-vectors are the average i-vectors calculated on the sentences from the training and validation data sets. Please note that these sentences are not necessarily parallelled. The source spectral feature and the source and target i-vectors are fed into the DBLSTM model, resulting in the target spectral feature. The vocoder finally re-synthesizes the target speech through the predicted spectral feature.

IV. Experimental Setup

A. I-vector Extractor

To extract reasonable i-vectors, we need speech data from many speakers to train UBM, TVS and LDA models. To this end, we use four corpus, including the wall street journal corpora (WSJ0+WSJ1) [26], British English data (WSJCAM) [27] and the voice cloning toolkit (VCTK)[4] corpus. Speech data are down-sampled to 16kHz and there are 647 speakers in total. I-vectors are extracted from gender-dependent GMM-UBMs. In the model training, 19-dimensional mel-frequency cepstral coefficients (MFCCs) and log-energy, with corresponding delta and delta-delta coefficients are extracted, and the window size is 25ms with a frame shift of 10ms. We use a simple energy-based voice activity detector (VAD) to remove silence frames before modeling training. The GMM has 512 Gaussian components and we calculate the sufficient statistics from UBM for every 10 sentences, which are used to extract one 400-dimensional i-vector. After LDA, we finally obtain the 17-dimensional i-vector together with one dimension of gender information. Each utterance of one specific speaker will generate a corresponding i-vector, and then we take the average of all the individual i-vectors for the speaker to form a single i-vector for identity representation. In our study, we use MSR identity toolbox [28] to extract i-vectors.

B. AVM Training and Voice Conversion

We use the VCTK corpus for voice conversion experiments, which contains speech data from 109 speakers, including 62 female and 47 male speakers. Each speaker originally has about 400 utterances, but we finally pick up about 30 parallel utterances between each inter-gender or intra-gender speaker pair. We aim to build inter-gender and intra-gender conversion

[3] By super-vector, we mean the dimention of the vector is quite large, e.g., 512 if 512 Gaussian mixtures are used in the UBM.

[4] http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html

TABLE I
THE NUMBER OF SPEAKERS AND THE NUMBER OF UTTERANCES IN TRAINING AND VALIDATION SETS FOR AVM TRAINING.

Conversion	Female-Male	Female-Female	Male-Female	Male-Male
Source speakers	59	27	43	21
Target speakers	44	29	59	22
Utterances in training set	4063	4923	4828	4902
Utterances in validation set	748	748	748	748

AVM with i-vectors. The number of source and target speakers and the number of utterances in the training and validation sets for AVM training are shown in TABLE I. Please note that Female-Male, Female-Female, Male-Female, Male-Male stand for conversions between female to male, female to female, male to female, and male to male respectively.

DTW is used to align the parallel data. STRAIGHT is used to extracted 50-dimensional mel-cepstral coefficients (MCCs), 513-dimensional aperiodic component and F_0. The 16KHz acoustic signal is windowed by 25ms and the frame shift is 5ms. The 49-dimensional (except for the energy dimension) MCCs are used in the DBLSTM conversions, while $LogF_0$ is linearly converted by equalizing the mean and the standard deviation of the source and target speech, and the aperiodic component of the source speech is directly copied to synthesize the converted speech. The training and validation samples are normalized to zero mean and unit variance for each dimension before DBLSTM training. A C++ CUDA-enabled library named CURRENNT[5] [29] is used to train the DBLSTM models with a learning rate of $1.0 * 10^{-5}$.

We conduct experiments for inter-gender and intra-gender voice conversion from a source speaker to a target speaker who are not included in the speakers for average voice model training. Meanwhile, 10 utterances from the source are selected as the testing data.

C. VC Systems

We implement four systems for experimentation:

- **DBLSTM**: The baseline approach is depicted in Fig. 1. We use 16 sentences as training data and 2 sentences as validation data. The input and output of the network are both spectral features (49-dimensional MCCs), and the number of units in each layer is [49 96 128 96 49].
- **DBSLTM+AVM**: We use parallel utterances from multiple speakers in the training set to train a source-to-target (e.g., female-to-male, female-to-female) average conversion model. The input/output and network setting are the same with the DBLSTM system.
- **DBLSTM+RM**: We retrain the DBLSTM+AVM model using some paralleled data from the testing source-target speaker pair (e.g., $p362 \rightarrow p227$). A set of 10 sentences are used for model retraining and 2 sentences for validation. We call this model as retrained model (RM).

TABLE II
THE MCD OF DIFFERENT VOICE CONVERSION SYSTEMS FOR INTER-GENDER AND INTRA-GENDER CONVERSION.

Conversion	Female-Male	Female-Female	Male-Female	Male-Male
Testing speaker pair	$p362 \rightarrow p227$	$p362 \rightarrow p351$	$p226 \rightarrow p351$	$p226 \rightarrow p227$
Source-Target	8.121	7.073	8.140	7.376
DBLSTM	6.515	5.725	6.015	6.001
DBSLTM+RM	6.081	5.216	5.487	5.582
DBSLTM+AVM+I-vector	6.628	5.774	5.868	6.073
DBLSTM+AVM	6.410	5.688	5.879	6.085

- **DBSLTM+AVM+I-vector**: The proposed approach is shown in Fig. 2. The dimension of the input feature is 83 (49-dimensional MCCs + 17-dimensional source i-vector + 17-dimensional target i-vector), while the output feature is the 49-dimensional MCCs of the target speech. The number of units in each layer in the DBLSTM is [83 96 128 96 49].

V. OBJECTIVE RESULTS

We use mel-cepstral distortion (MCD) to objectively measure the spectral distortion between the converted and the target speech [7] [11]:

$$\text{MCD}[dB] = \frac{10}{ln10}\sqrt{2\sum_{d=1}^{N}(C_d - C_d^{con})^2} \tag{3}$$

where C_d and C_d^{con} are the d-th coefficient of the target and converted MCCs, respectively; N is the dimension of MCCs (except for energy dimension). Lower MCD means smaller distortion.

The MCD scores of the four voice conversion systems for inter-gender and intra-gender conversion are shown in TABLE II. We note that the lowest MCD is achieved by DBLSTM+RM. This means retraining using source-target paralleled data on an average voice model is beneficial. It is surprising that, serving as an *average* model, DBLSTM+AVM achieves even lower MCD than the direct conversion model – DBLSTM. This is probable because the training data for direct conversion DBLSTM is quite limited. However, in real applications, we cannot have many utterances for model training. We also notice that the AVM with augmented i-vectors (DBLSTM+AVM+I-vector) keeps almost the same or increases the MCD a little bit as compared with DBSLTM+AVM.

VI. SUBJECTIVE RESULTS

We conduct subjective listening tests[6] for Female-Male conversion to compare the four systems (DBLSTM, DBLSTM+RM, DBLSTM+AVM+I-vector and DBLSTM+AVM). We carry out AB and ABX preference tests for quality and similarity respectively on three pairs: DBLSTM+RM vs. DBLSTM, DBLSTM+AVM+I-vector vs. DBLSTM+RM, and

[5] https://sourceforge.net/projects/currennt/

[6] Alternatively, an automatic voice conversion evaluation strategy [30] may be used.

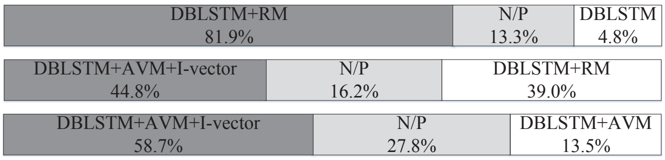


Fig. 4. AB preference test for speech quality. The p-values of the three pairs are $6.69 * 10^{-19}$, 0.24, and $1.39 * 10^{-8}$, respectively.

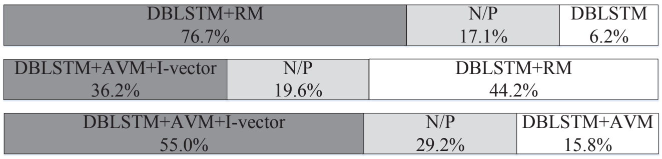


Fig. 5. ABX preference test for speaker similarity. The p-values of the three pairs are $1.23 * 10^{-17}$, 0.21 and $4.31 * 10^{-8}$, respectively.

DBLSTM+AVM+I-vector vs. DBLSTM+AVM. We recruit 20 listeners to evaluate 10 sentences, resulting in 200 votes for each system. The quality and similarity preference bars are shown in Fig. 4 and Fig. 5, respectively.

A. DBLSTM+RM vs. DBLSTM

As a sanity check, we would like to see if retraining using source-target paralleled data on an average voice model is beneficial (as we can see an MCD decrease in objective evaluation). From Fig. 4 and Fig. 5, it is obvious that DBLSTM+RM achieves significantly better preference than DBLSTM in both quality and similarity (confirmed by one-way ANOVA analysis of variance). This confirms that an average voice model is quite useful to reduce the amount of parallel training data and to improve the performance of the converted speech.

B. DBLSTM+AVM+I-vector vs. DBLSTM+RM

From the second bar in Fig. 4 and Fig. 5, we can see that DBLSTM+AVM+I-vector achieves a little bit better preference than DBLSTM+RM in quality but with opposite observation in similarity. One-way ANOVA analysis of variance shows that the differences between the two systems are not significant both in quality and similarity. This means that the DBLSTM+AVM+I-vector approach can achieve almost equivalent performance with DBLSTM+RM that needs parallel data to retain an average model.

C. DBLSTM+AVM+I-vector vs. DBLSTM+AVM

Finally, as another sanity check, we want to see if an AVM without i-vectors can perform equally with an AVM with i-vectors. From the third bar in the two figures (Fig. 4 and Fig. 5), we can see that the DBLSTM+AVM+I-vector achieves significant better performance than DBLSTM+AVM both in quality and similarity (the differences are significant). This indicates that the average voice model without i-vector information cannot capture the characteristics of a specific speaker, leading to poor quality and similarity in the converted speech. On the contrary, using i-vectors can capture the speaker identity and achieve high quality at the same time.

We also notice that the subjective results are not quite consistent with the objective results. This is understandable because objective scores might not always be well consistent with human subjective perception, and it only provides a practical and effective way to optimize the systems, especially for tuning hyper-parameters [31].

VII. Conclusions

We propose to use average voice model (AVM) with i-vectors in DBLSTM based voice conversion framework, which does not require parallel data between source and target speaker for conversion. Specifically in our approach, an AVM is trained by many other speakers' parallel data. I-vectors, which represent the identities of source and target speakers, are augmented with source speech as the DBLSTM input. Without an adaptation stage, the AVM+i-vector model can be directly used for voice conversion: given the spectrum of a new source speaker and his/her i-vector, along with the i-vector of the new target speaker, the network directly generates the corresponding spectrum for the new target speaker. Our study shows that the proposed approach is quite effective. In conclusion, our approach can make voice conversion greatly convenient and flexible in real applications, meanwhile it guarantees reasonable quality and similarity of converted speech. Some samples used in the listening test are available via this link: http://www.nwpu-aslp.org/vc/apsipa-jiewu-demo.pptx.

Acknowledgments

This paper is partially supported by a grant from National Science Foundation of China (No.61571363).

References

[1] T. Toda, D. Saito, F. Villavicencio, J. Yamagishi, M. Wester, Z. Wu, L.-H. Chen *et al.*, "The voice conversion challenge 2016," in *INTERSPEECH*, 2016.

[2] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results," in *INTERSPEECH*, 2016.

[3] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, vol. 1. IEEE, 1998, pp. 285–288.

[4] C. Veaux and X. Rodet, "Intonation conversion from neutral to expressive speech." in *INTERSPEECH*, 2011, pp. 2765–2768.

[5] D. Hironori, K. Nakamura, T. Tomoki, H. Saruwatari, and K. Shikano, "Esophageal speech enhancement based on statistical voice conversion with gaussian mixture models," *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 9, pp. 2472–2482, 2010.

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

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

[8] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 3, pp. 806–817, 2012.

[9] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 313–317.

[10] H. Ming, D. Huang, L. Xie, S. Zhang, M. Dong, and H. Li, "Exemplar-based sparse representation of timbre and prosody for voice conversion," in *ICASSP*. IEEE, 2016, pp. 5175–5179.

[11] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, “Voice conversion using artificial neural networks,” in *ICASSP*. IEEE, 2009, pp. 3893–3896.

[12] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, “Voice conversion in high-order eigen space using deep belief nets.” in *INTERSPEECH*, 2013, pp. 369–372.

[13] 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 Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[14] L. Sun, S. Kang, K. Li, and H. Meng, “Voice conversion using deep bidirectional long short-term memory based recurrent neural networks,” in *ICASSP*. IEEE, 2015, pp. 4869–4873.

[15] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, “Voice conversion with smoothed gmm and map adaptation,” in *Eurospeech-2003*, 2003, pp. 2413–2416.

[16] C.-H. Lee and C.-H. Wu, “Map-based adaptation for speech conversion using adaptation data selection and non-parallel training.” in *INTERSPEECH*, 2006.

[17] T. Toda, Y. Ohtani, and K. Shikano, “Eigenvoice conversion based on gaussian mixture model,” in *INTERSPEECH*, 2006.

[18] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000.

[19] Y. Ohtani, T. Tomoki, H. Saruwatari, and K. Shikano, “Adaptive training for voice conversion based on eigenvoices,” *IEICE transactions on information and systems*, vol. 93, no. 6, pp. 1589–1598, 2010.

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

[21] H. Kawahara, “Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1303–1306.

[22] Z. Wu, O. Watts, and S. King, “Merlin: An open source neural network speech synthesis system,” in *9th ISCA Speech Synthesis Workshop (SSW9)*, 2016.

[23] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, “A study of speaker adaptation for dnn-based speech synthesis,” in *INTERSPEECH*, 2015.

[24] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[25] R. Vogt, S. S. Kajarekar, and S. Sridharan, “Discriminant nap for svm speaker recognition,” in *ODYSSEY*, 2008.

[26] D. B. Paul and J. M. Baker, “The design for the wall street journal-based csr corpus,” in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[27] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, “Wsjcamo: a british english speech corpus for large vocabulary continuous speech recognition,” in *ICASSP*, vol. 1. IEEE, 1995, pp. 81–84.

[28] S. O. Sadjadi, M. Slaney, and L. Heck, “Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research,” *Speech and Language Processing Technical Committee Newsletter*, vol. 1, no. 4, 2013.

[29] F. Weninger, J. Bergmann, and B. Schuller, “Introducing currennt–the munich open-source cuda recurrent neural network toolkit,” *Journal of Machine Learning Research*, vol. 16, no. 3, pp. 547–551, 2015.

[30] D. Huang, L. Xie, Y. Lee, J. Wu, H. Ming, X. Tian, S. Zhang, C. Ding, M. Li, Q. Nguyen *et al.*, “An automatic voice conversion evaluation strategy based on perceptual background noise distortion and speaker similarity,” in *9th ISCA Speech Synthesis Workshop (SSW9)*, 2016.

[31] Z. Wu and S. King, “Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum generation error training,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1255–1265, 2016.