SPARSE NONLINEAR REPRESENTATION FOR VOICE CONVERSION

Toru Nakashika

University of Electro-Communications
Graduate School of Information Systems
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
nakashika@is.uec.ac.jp

Tetsuya Takiguchi, Yasuo Ariki

Kobe University
Graduate School of System Informatics
1-1 Rokkodai, Kobe, Japan
takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

ABSTRACT

In voice conversion, sparse-representation-based methods have recently been garnering attention because they are, relatively speaking, not affected by over-fitting or over-smoothing problems. In these approaches, voice conversion is achieved by estimating a sparse vector that determines which dictionaries of the target speaker should be used, calculated from the matching of the input vector and dictionaries of the source speaker. The sparse-representation-based voice conversion methods can be broadly divided into two approaches: 1) an approach that uses raw acoustic features in the training data as parallel dictionaries, and 2) an approach that trains parallel dictionaries from the training data. In our approach, we follow the latter approach and systematically estimate the parallel dictionaries using a joint-density restricted Boltzmann machine with sparse constraints. Through voice-conversion experiments, we confirmed the high-performance of our method, comparing it with the conventional Gaussian mixture model (GMM)-based approach, and a non-negative matrix factorization (NMF)-based approach, which is based on sparse representation.

Index Terms— Voice Conversion, Restricted Boltzmann Machine, Joint Density, Sparse Representation, Parallel Dictionary Learning

1. INTRODUCTION

Voice conversion (VC) is a technique that changes specific information in the speech of a source speaker to that of a target speaker while retaining linguistic information. This technique has been applied to various tasks, such as speech enhancement [1], emotion conversion [2], speaking assistance [3], and other applications [4, 5]. The most important role of VC is that one can analyze and control the speaker identity in a speech signal, and hence VC has been garnering attention in multimedia signal processing as a fundamental tool [6].

Various statistical approaches to VC have been studied so far, including those discussed in [7, 8]. Among these approaches, the joint density Gaussian mixture model (JD-GMM)-based mapping method [9] is widely used, and a number of improvements have been proposed [10, 11, 12, 13]. However, it is reported that the GMM-based approaches have some problems related to over-fitting and over-smoothing [14, 15, 16]. The problems come from the linear conversion that aggregates adjacent data points to each Gaussian mode. Sparse-representation-based approaches using non-negative matrix factorization (NMF) [17] or unit selection (US) [15] have, therefore, been proposed to alleviate such problems.

In sparse-representation-based VC, one obtains the target speaker's speech by estimating a sparse vector that determines which dictionaries should be used. The sparse vector is calculated by matching the source speaker's input vector against the dictionaries. Therefore, the pairs of dictionaries (called parallel dictionaries) should be trained beforehand. In this approach, the converted speech quality mainly depends on the accuracy when creating the parallel dictionary and when selecting the appropriate dictionaries (estimating a sparse vector). Exemplar-based VC using NMF [18] uses the spectra in the parallel training data as dictionaries. Therefore, although it is not affected by the errors that occur when creating dictionaries, it degrades conversion accuracy by the errors that occur when estimating sparse vectors (a mismatch of activity matrices of the source speaker and the target speaker). Another NMF-based VC method, in which parallel dictionaries are trained from the training data so as to match their activity matrices instead of using the exemplars as it is, has been also proposed [19]. This approach decreases the errors that occur by mismatching the activity matrices; however, it produces errors in the training of parallel dictionaries in contrast.

The above-mentioned VC methods are based on linear functions. Since our vocal tracts have non-linear shapes, a non-linear function is a better representation for capturing non-linear relationships between a source speaker's speech and target speaker's speech. Several voice conversion methods based on non-linear functions can be found in [20] by Desai *et al.* (they employ multi-layer neural networks (NNs)), in [21] by Ling *et al.* (they use a restricted Boltzmann machine (RBM)), and in [22] by Wu *et al.* (they use a conditional restricted Boltzmann machine (CRBM [23])). Nakashika *et al.* also employed another non-linear approach that uses speaker-

dependent RBMs or CRBMs to capture speaker-specific features [24, 25]. It has been reported that these graphical models are better at representing the distribution of high-dimensional observations with cross-dimension correlations than GMM in speech synthesis [14] and in speech recognition [26]. Since Hinton et al. introduced an effective training algorithm in 2006 [27], the use of deep learning rapidly spread in the field of multimedia signal processing [27, 28, 29]

In this paper, we describe a non-linear voice conversion method that utilizes a joint density RBM with sparse constraints in order to effectively train parallel dictionaries of source/target speakers in the framework of a sparse-representation approach. An RBM is a bi-directional probabilistic model that consists of a visible layer and a hidden layer, characterized in that there is no connection among the units in the same layer, but there exist connections among the units in different layers. These connection weights (parallel dictionaries) are trained in an unsupervised manner. In our approach, an RBM inputs a concatenated vector (parallel data) of the source speaker's and the target speaker's acoustic features, such as MFCC, that are aligned beforehand. By feeding such vectors, the RBM trains *co-occurrences* of the source speaker's features and the target speaker's features through hidden units. Our approach is similar to Ling's work [21]. While Ling's approach alternatively used an RBM instead of using GMM to capture the joint distribution, our approach tries to train parallel dictionaries in sparse-representation-based voice conversion. Nakashika's works [24, 25] also used RBMs (or deeper architectures) for each speaker to extract speaker-specific features, while our approach feeds a concatenated vector of the speakers in the visible layer.

2. SPARSE REPRESENTATION FOR VC

In voice conversion, the system generally converts an acoustic vector of the source speaker $\boldsymbol{x} \in \mathbb{R}^D$ into the target speaker's vector $\boldsymbol{y} \in \mathbb{R}^D$. In sparse-representation-based voice conversion, K pairs of the source speaker's dictionaries $\mathcal{D}_x \in \mathbb{R}^{D\times K}$ and the target speaker's dictionaries $\mathcal{D}_y \in \mathbb{R}^{D\times K}$ (parallel dictionaries) are trained beforehand. Given an input vector of the source speaker, the converted target speaker's vector $\boldsymbol{y}$ is obtained using the trained parallel dictionaries. First, we calculate a sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^K, \|\boldsymbol{\alpha}\|_0 \ll K$ that satisfies

$$\boldsymbol{x} \approx f(\mathcal{D}_x\boldsymbol{\alpha}), \tag{1}$$

and then we obtain the target speaker's vector as follows:

$$\boldsymbol{y} \approx f(\mathcal{D}_y\boldsymbol{\alpha}), \tag{2}$$

where $f(\cdot)$ indicates an arbitrary gate function.

Most sparse-representation-based approaches use training exemplars without changes for the parallel dictionaries $\mathcal{D}_x, \mathcal{D}_y$ [30, 31, 18]. For the calculation of the sparse vector $\boldsymbol{\alpha}$, there are various approaches that can be used, such as L1 normalization [30], K-nearest neighbors algorithm [31], and sparse non-negative matrix factorization (NMF) [18]. In [19], another approach using sparse NMF has been proposed that uses trained parallel dictionaries so that the sparse vectors for the source and the target are the same. Furthermore, the above-mentioned sparse-representation-based voice conversion methods are based on linear function ($f(\cdot)$ are linear functions in Eqs. (1) and (2)).

3. PRELIMINARY

3.1. Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is an undirected graphical model that defines the distribution of visible variables with binary hidden (latent) variables [32]. In literature dealing with a Gaussian-Bernoulli RBM (GBRBM [33]), the joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ of real-valued visible units $\boldsymbol{v} = [v_1, \cdots, v_I]^T, v_i \in \mathbb{R}$ and binary-valued hidden units $\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0, 1\}$ are defined as follows:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{3}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = \left\| \frac{\boldsymbol{v} - \boldsymbol{b}}{2\boldsymbol{\sigma}} \right\|^2 - \boldsymbol{c}^T\boldsymbol{h} - \left(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2}\right)^T \boldsymbol{W}\boldsymbol{h} \tag{4}$$

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}, \tag{5}$$

where $\|\cdot\|^2$ denotes L2 norm. $\boldsymbol{W} \in \mathbb{R}^{I\times J}$, $\boldsymbol{\sigma} \in \mathbb{R}^{I\times 1}$, $\boldsymbol{b} \in \mathbb{R}^{I\times 1}$, and $\boldsymbol{c} \in \mathbb{R}^{J\times 1}$ are model parameters of the GBRBM, indicating the weight matrix between visible units and hidden units, the standard deviations associated with Gaussian visible units, a bias vector of the visible units, and a bias vector of hidden units, respectively. The fraction bar in Eq. (4) denotes the element-wise division.

Because there are no connections between visible units or between hidden units, the conditional probabilities $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ form simple equations as follows:

$$p(h_j = 1|\boldsymbol{v}) = \mathcal{S}\left(c_j + \left(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2}\right)^T \boldsymbol{W}_{:j}\right) \tag{6}$$

$$p(v_i = v|\boldsymbol{h}) = \mathcal{N}\left(v \mid b_i + \boldsymbol{W}_{i:}\boldsymbol{h}, \sigma_i^2\right), \tag{7}$$

where $\boldsymbol{W}_{:j}$ and $\boldsymbol{W}_{i:}$ denote the jth column vector and the ith row vector, respectively. $\mathcal{S}(\cdot)$ and $\mathcal{N}(\cdot|\mu, \sigma^2)$ indicate an element-wise sigmoid function and Gaussian probability density function with the mean μ and variance σ^2.

For parameter estimation, the following negative log-likelihood of a collection of visible units is used as an evaluation function.

$$\mathcal{L}_{RBM} = -\log \prod_n p(\boldsymbol{v}^n) = -\sum_n \log \sum_{\boldsymbol{h}} p(\boldsymbol{v}^n, \boldsymbol{h}^{\boldsymbol{n}}) \tag{8}$$

However, it is generally difficult to compute the exact gradient, contrastive divergence (CD) is used instead [27].

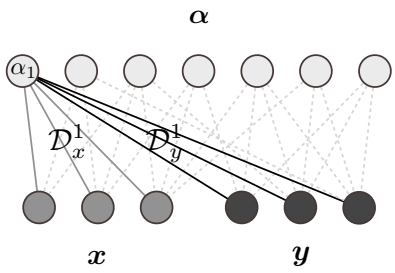

Fig. 1. A joint density RBM for voice conversion. Dictionaries of the source and the target speakers are connected to each other via a hidden sparse vector $\boldsymbol{\alpha}$.

Input: Dictionaries $\mathcal{D}_x$, $\mathcal{D}_y$, a source speaker's vector $\boldsymbol{x}$ and an initial target vector $\boldsymbol{y}_0$
Output: Estimated target speaker's vector $\hat{\boldsymbol{y}}$
Initialize: Set the initial values as $\hat{\boldsymbol{y}} = \boldsymbol{y}_0$.

Repeat the following updates R times:

1. $\hat{\boldsymbol{\alpha}} \triangleq \mathbb{E}[\boldsymbol{\alpha}]_{p(\boldsymbol{\alpha}|\boldsymbol{x},\hat{\boldsymbol{y}})} = \mathcal{S}(\mathcal{D}_x^T(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_x^2}) + \mathcal{D}_y^T(\frac{\hat{\boldsymbol{y}}}{\boldsymbol{\sigma}_y^2}) + \boldsymbol{c})$
2. $\hat{\boldsymbol{y}} \triangleq \mathbb{E}[\boldsymbol{\alpha}]_{p(\boldsymbol{y}|\hat{\boldsymbol{\alpha}})} = \mathcal{D}_y\hat{\boldsymbol{\alpha}} + \boldsymbol{b}_y$

Fig. 2. Iterative estimation algorithm of the target vector using a joint density RBM.

3.2. RBM with sparse constraints

Essentially, an RBM itself is capable of making hidden units sparse, since there are no connections in the hidden layer. However, if the number of the hidden units is not enough, they sometimes turn to be non-sparse. In our approach, we employ sparse constraints as in [34] to make the hidden units more sparse. Introducing the sparse constraints, each parameter is optimized so as to minimize the total cost $\mathcal{L}_{RBM} + \lambda\mathcal{L}_{sp}$, where λ is a hyper parameter that determines the strength of the sparsity. The regularization term $\mathcal{L}_{sp}$ is defined as:

$$\mathcal{L}_{sp} = \sum_j |p - \frac{1}{N}\sum_n \mathbb{E}[h_j^n|\boldsymbol{v}^n]|^2, \tag{9}$$

where N is the number of training data, and p is a constant that controlls the sparseness (typically, $p = 0.05$ is used). As Eq. (9) indicates, this regularization makes the average value of h_j given $\boldsymbol{v}$ close to the small value of p; consequently, $\boldsymbol{h}$ becomes sparse.

4. PARALLEL DICTIONARY LEARNING USING A JOINT DENSITY MODEL

Our voice conversion system uses a joint density RBM with sparse constraints to train parallel dictionaries $\mathcal{D}_x$, $\mathcal{D}_y$ and estimate sparse vectors $\boldsymbol{\alpha}$ at the same time as shown in Fig. 1. As discussed in the previous section, an RBM is a two-layer network that consists of a visible layer and a hidden layer, characterized in that bi-directional connections exist only between visible and hidden units. As shown in Fig. 1, the RBM that feeds a concatenated vector of source speaker's features $\boldsymbol{x}$ and target speaker's features $\boldsymbol{y}$ can be regarded as a network where a dictionary-selection weight (a sparse vector) $\boldsymbol{\alpha}_i$ connects to both $\boldsymbol{x}$ and $\boldsymbol{y}$ with weights of ith dictionaries $\mathcal{D}_x^i$ and $\mathcal{D}_y^i$, respectively.

Given parallel training data $(\boldsymbol{x}, \boldsymbol{y})$, we define a joint probability of $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\alpha}$ as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\alpha}; \mathcal{D}_x, \mathcal{D}_y) = \frac{1}{Z}e^{-E(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\alpha};\mathcal{D}_x,\mathcal{D}_y)} \tag{10}$$

$$E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\alpha}; \mathcal{D}_x, \mathcal{D}_y) = \left\|\frac{\boldsymbol{x} - \boldsymbol{b}_x}{2\boldsymbol{\sigma}_x}\right\|^2 + \left\|\frac{\boldsymbol{y} - \boldsymbol{b}_y}{2\boldsymbol{\sigma}_y}\right\|^2 - \boldsymbol{c}^T\boldsymbol{\alpha} - \left(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_x^2}\right)^T\mathcal{D}_x\boldsymbol{\alpha} - \left(\frac{\boldsymbol{y}}{\boldsymbol{\sigma}_y^2}\right)^T\mathcal{D}_y\boldsymbol{\alpha} \tag{11}$$

where $Z = \sum_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\alpha}} e^{-E(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\alpha})}$ indicates a normalization term, and $\boldsymbol{c}$ is a bias parameter vector of a dictionary-selection weights. $\boldsymbol{b}_x$ and $\boldsymbol{\sigma}_x$ indicate bias and deviation parameters of the source speaker's acoustic features, respectively, and $\boldsymbol{b}_y$ and $\boldsymbol{\sigma}_y$ indicate bias and deviation parameters of the target speaker's features, respectively. The dictionaries $\mathcal{D}_x, \mathcal{D}_y$ (and the other parameters) can be estimated by minimizing the cost function $\mathcal{L} = \mathcal{L}_{JDRBM} + \lambda\mathcal{L}_{sp}$, where

$$\begin{aligned}\mathcal{L}_{JDRBM} &= -\log\prod_n p(\boldsymbol{x}^n, \boldsymbol{y}^n) \\ &= -\sum_n \log\sum_{\boldsymbol{\alpha}} p(\boldsymbol{x}^n, \boldsymbol{y}^n, \boldsymbol{\alpha}; \mathcal{D}_x, \mathcal{D}_y).\end{aligned} \tag{12}$$

As discussed in Section 3, we can make practical use of an approximation method (contrastive divergence) to calculate the gradients.

When it comes to conversion, we estimate the target speaker's vector $\hat{\boldsymbol{y}}$ by repeating forward inference and backward inference of an RBM as shown in Fig. 2. Given an initial vector $\boldsymbol{y}_0$, we first calculate the expectation values of dictionary-selection weights $\boldsymbol{\alpha}$ using the probability that each dictionary is selected:

$$p(\boldsymbol{\alpha} = \boldsymbol{1}|\boldsymbol{x}, \boldsymbol{y}) = \mathcal{S}(\mathcal{D}_x^T(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_x^2}) + \mathcal{D}_y^T(\frac{\boldsymbol{y}}{\boldsymbol{\sigma}_y^2}) + \boldsymbol{c}). \tag{13}$$

Secondly, the expectation values of $\boldsymbol{y}$ are calculated using $\hat{\boldsymbol{\alpha}}$. The conditional probability of $\boldsymbol{y}$ is given from backward inference of an RBM as follows:

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{y}|\mathcal{D}_y\boldsymbol{\alpha} + \boldsymbol{b}_y, \boldsymbol{\sigma}_y^2). \tag{14}$$

Repeating the above-mentioned procedures (estimation of $\boldsymbol{\alpha}$ and $\boldsymbol{y}$) R times, we iteratively obtain the converted vector $\hat{\boldsymbol{y}}$. Although several approaches for determining the initial vector $\boldsymbol{y}_0$ can be considered, we use the source feature vector $\boldsymbol{x}$ for the initial values in this paper.

Similar to SMNMF (spectral mapping NMF [19]), our voice conversion method optimizes the likelihood of the training data as well as the likelihood of the dictionary-selection vector as shown in Eq. (10). The most obvious difference is that our approach uses a non-linear function for estimating a dictionary-selection vector as in Eq. (13), while SMNMF still uses a linear function. Furthermore, while SMNMF is restricted to input non-negative values, our approach can feed real values without constraints. In particular, MFCC, which tends to distribute monomodally, will go together with our approach that assumes Gaussian-distributed inputs.

5. EXPERIMENTS

5.1. Conditions

In our experiments, we conducted voice conversion using the ATR Japanese speech database [35], comparing our method (joint density restricted Boltzmann machines with sparse constraints, or "JDRBM+s") with the conventional sparse-representation-based voice conversion that uses exemplar-based NMF [18], and spectral-mapping-based NMF [19] and, for a reference, the well-known GMM-based approach (64 mixtures). From this database, we used a male speaker (identified with "MMY") and a female speaker ("FTK") for the source and target speakers, respectively. As an acoustic feature vector for our approach and GMM, we calculated 24-dimensional MFCC features from STRAIGHT spectra [36] using filter-theory [37] to decode the MFCC back to STRAIGHT spectra in the synthesis stage. For NMF-based approaches, we used 513-dimensional vectors of STRAIGHT spectra. The parallel data of the source/target speakers processed by Dynamic Programming were created from 216 word utterances (58,426 frames) in the dataset, and were used for the training of each method. For the objective test, 25 sentences (about 100 sec. long) that were not included in the training data were arbitrarily selected from the database. The joint density RBM was trained using gradient descent with a learning rate of 0.01 and momentum of 0.9, with the number of epochs being 200. We set the number of hidden units as 96. We changed the sparse-constraint strength λ as 0, 1, 10, 100, and evaluated their performance. We obtained the converted vector with $R = 10$ iterations (already converged). For spectral-mapping-based NMF, we changed the number of bases k as 1,000 and 2,500. For exemplar-based NMF, we compared the case where all training frames were used ($k = 58426$) and the case where 1,000 frames were arbitrarily used from the training data ($k = 1000$).

For the objective evaluation, we used SDIR (spectral distortion improvement ratio) to measure how the converted vector is improved to resemble the original source vector. The SDIR is defined as follows:

$$SDIR[dB] = 10\log_{10}\frac{\sum_d |\boldsymbol{X}^t(d) - \boldsymbol{X}^s(d)|^2}{\sum_d |\boldsymbol{X}^t(d) - \hat{\boldsymbol{X}}^t(d)|^2}, \qquad (15)$$

where $\boldsymbol{X}^t(d)$, $\boldsymbol{X}^s(d)$ and $\hat{\boldsymbol{X}}^t(d)$ denote the dth original target spectra, the source spectra and the converted spectra (spectra obtained from the converted MFCC), respectively. The larger the value of SDIR is, the greater the improvement in the converted spectra. We calculated the SDIR for each frame in the training data, and averaged the SDIR values for the final evaluation.

5.2. Results and discussion

We summarize the experimental results of each method in Table 1. As shown in Table 1, our approach outperformed the other methods ($\lambda = 10$ performed best). The differences between our approach and the NMF-based approach are the types of input features and the gate functions in Eqs. (1) and (2). The reason for the improvement is attributed to the fact that our approach, which inputs real-valued data and uses non-linear gate functions, is able to represent input features better than the NMF-based approach. We obtained better results as the strength of sparsity increased, although the performance degrades when we make the hidden units too sparse ($\lambda = 100$). This is becasue if we make the hidden units sparse, the obtained target vector $\hat{\boldsymbol{y}}$ becomes more clearer and improved without having scrambled by a lot of dictionaries. Meanwhile, however, if we make the hidden units too sparse, this not only makes it difficult to estimate dictionaries precisely, but also vanishes activities of the dictionary selection in the conversion step (as shown in the bottom right in Fig. 3). Fig. 3 shows an exmaple of expectation values of the estimated hidden units. As shown in Fig. 3, the hidden units gradually become sparse (almost all of the values in $\boldsymbol{\alpha}$ are zero) as the strength λ increases. One interesting point is that even if we do not give any sparse constraints (i.e., $\lambda = 0$), the hidden units are already sparse to some extent. This is due to the fact that an RBM characteristically naturally makes the hidden units sparse in the process where the model parameters are estimated so that the hidden units do not capture redundant information between each other.

6. CONCLUSION

This paper presented a VC method using a joint density RBM with sparse constraints as an alternative tool of a sparse-representation-based approach where only a few dictionaries are used for the converted-voice generation. Our proposed method demonstrated better performance compared with the conventional sparse-representation-based approach (spectral-mapping NMF and exemplar-based NMF) and the

Table 1. Performance of each method.

	JDRBM+s (Proposed)				Spectral mapping NMF		Exemplar-based NMF		GMM
Methods	$\lambda = 0$	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$	$k = 1000$	$k = 2500$	$k = 1000$	$k = 58426$	$m = 64$
SDIR [dB]	4.96	5.54	**6.00**	5.04	5.14	4.68	4.91	5.23	4.11

Fig. 3. Expectation values of the estimated $\boldsymbol{\alpha}$ from a part of a sentence when the strength of the sparsity changed as $\lambda = 0, 1, 10, 100$. The vertical and horizontal axes indicate the index of hidden units and the time, respectively.

well-known GMM-based approach. When we compare the results between our approach and the spectral-mapping NMF, we could say that the non-linear conversion function without non-negative constraints plays an important role in sparse-representation-based VC. In the future, we will extend our method to have a deeper architecture using a deep Boltzmann machine or such, so that it captures more complex information in the data.

7. REFERENCES

[1] Alexander Kain and Michael W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, 1998, pp. 285–288.

[2] Christophe Veaux and X. Robet, "Intonation conversion from neutral to expressive speech," in *Interspeech*, 2011, pp. 2765–2768.

[3] Keigo Nakamura, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[4] Li Deng, Alex Acero, Li Jiang, Jasha Droppo, and Xuedong Huang, "High-performance robust speech recognition using stereo training data," in *ICASSP*, 2001, pp. 301–304.

[5] Aki Kunikoshi, Yu Qiao, Nobuaki Minematsu, and Keikichi Hirose, "Speech generation from hand gestures based on space mapping," in *Interspeech*, 2009, pp. 308–311.

[6] Kenta Masaka, Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki, "Multimodal exemplar-based voice conversion using lip features in noisy environments," in *Interspeech*, 2014, pp. 1159–1163.

[7] Robert Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[8] H. Valbret, E. Moulines, and Jean-Pierre Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, no. 2, pp. 175–187, 1992.

[9] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech, Audio Process.*, vol. 6, no. 2, pp. 131–142, 1998.

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

[11] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj, "Voice conversion using partial least squares regression," *IEEE Trans. Speech, Audio Process.*, vol. 18, no. 5, pp. 912–921, 2010.

[12] Daisuke Saito, Keisuke Yamamoto, Nobuaki Minematsu, and Keikichi Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Interspeech*, 2011, pp. 653–656.

[13] Chung-Han Lee and Chung-Hsien Wu, "Map-based adaptation for speech conversion using adaptation data selection and non-parallel training," in *Interspeech*, 2006, pp. 2254–2257.

[14] Z-H Ling, Li Deng, and Dong Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE*

Trans. Audio, Speech, Lang. Process., , no. 10, pp. 2129–2139, 2013.

[15] Zhizheng Wu, Tuomas Virtanen, Tomi Kinnunen, Eng Siong Chng, and Haizhou Li, "Exemplar-based unit selection for voice conversion utilizing temporal information," in *Interspeech*, 2013, pp. 3057–3061.

[16] Hsin-Te Hwang, Yu Tsao, Hsin-Min Wang, Yih-Ru Wang, and Sin-Horng Chen, "Alleviating the over-smoothing problem in gmm-based voice conversion with discriminative training," in *Interspeech*, 2013, pp. 3062–3066.

[17] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2000, pp. 556–562.

[18] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Exemplar-based voice conversion in noisy environment," in *SLT*, 2012, pp. 313–317.

[19] Ryoichi Takashima, Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki, "Noise-robust voice conversion based on spectral mapping on sparse space," in *SSW8*, 2013, pp. 71–75.

[20] Srinivas Desai, E. Veera Raghavendra, B. Yegnanarayana, Alan W. Black, and Kishore Prahallad, "Voice conversion using artificial neural networks," in *ICASSP*. IEEE, 2009, pp. 3893–3896.

[21] Ling-Hui Chen, Zhen-Hua Ling, Yan Song, and Li-Rong Dai, "Joint spectral distribution modeling using restricted boltzmann machines for voice conversion," in *Interspeech*, 2013, pp. 3052–3056.

[22] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Conditional restricted boltzmann machine for voice conversion," in *ChinaSIP*, 2013.

[23] Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis, "Modeling human motion using binary latent variables," in *Advances in neural information processing systems*, 2006, pp. 1345–1352.

[24] Toru Nakashika, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *Interspeech*, 2013, pp. 369–372.

[25] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki, "Speaker-dependent conditionl restricted boltzmann machine for voice conversion," *IEICE Technical Report SP2013-88*, vol. 113, no. 366, pp. 83–88, 2013.

[26] Abdel-rahman Mohamed, George E. Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, 2012.

[27] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[28] Vinod Nair and Geoffrey E Hinton, "3D object recognition with deep belief nets.," in *NIPS*, 2009, pp. 1339–1347.

[29] Thomas Deselaers, Saša Hasan, Oliver Bender, and Hermann Ney, "A deep learning approach to machine transliteration," in *Statis. Machine Trans.*, 2009, pp. 233–241.

[30] David L Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[31] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition*. IEEE, 2004, vol. 1, pp. 275–282.

[32] Yoav Freund and David Haussler, *Unsupervised learning of distributions of binary vectors using two layer networks*, Computer Research Laboratory, 1994.

[33] KyungHyun Cho, Alexander Ilin, and Tapani Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *ICANN*, pp. 10–17. Springer, 2011.

[34] Honglak Lee, Chaitanya Ekanadham, and Andrew Y Ng, "Sparse deep belief net model for visual area v2," in *Advances in Neural Info. Process. Systems*, 2008, pp. 873–880.

[35] Akira Kurematsu, Kazuya Takeda, Yoshinori Sagisaka, Shigeru Katagiri, Hisao Kuwabara, and Kiyohiro Shikano, "ATR japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.

[36] Hideki Kawahara, Masanori Morise, Toru Takahashi, Ryuichi Nisimura, Toshio Irino, and Hideki Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *ICASSP*. IEEE, 2008, pp. 3933–3936.

[37] Ben Milner and Xu Shao, "Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model," in *Interspeech*, 2002, pp. 2421–2424.