# VOICE CONVERSION IN TIME-INVARIANT SPEAKER-INDEPENDENT SPACE

*Toru Nakashika, Tetsuya Takiguchi, Yasuo Ariki*

Graduate School of System Informatics, Kobe University
1-1 Rokkodai, Nada-ku, Kobe, 657-8501 Japan
nakashika@me.cs.scitec.kobe-u.ac.jp, {takigu,ariki}@kobe-u.ac.jp

## ABSTRACT

In this paper, we present a voice conversion (VC) method that utilizes conditional restricted Boltzmann machines (CRBMs) for each speaker to obtain time-invariant speaker-independent spaces where voice features are converted more easily than those in an original acoustic feature space. First, we train two CRBMs for a source and target speaker independently using speaker-dependent training data (without the need to parallelize the training data). Then, a small number of parallel data are fed into each CRBM and the high-order features produced by the CRBMs are used to train a concatenating neural network (NN) between the two CRBMs. Finally, the entire network (the two CRBMs and the NN) is fine-tuned using the acoustic parallel data. Through voice-conversion experiments, we confirmed the high performance of our method in terms of objective and subjective evaluations, comparing it with conventional GMM, NN, and speaker-dependent DBN approaches.

***Index Terms—*** Voice conversion, conditional restricted Boltzmann machine, deep learning, speaker specific features

## 1. INTRODUCTION

In recent years, voice conversion (VC), a technique used to change specific information in the speech of a source speaker to that of a target speaker while retaining linguistic information, has been garnering much attention in speech signal processing. VC techniques have been applied to various tasks, such as speech enhancement [1], emotion conversion [2], speaking assistance [3], and other applications [4, 5]. Most of the related work in VC focuses not on f0 conversion but on the conversion of spectrum features, and we conform to that in this report as well.

Various statistical approaches to VC have been studied so far, for example those discussed in [6, 7]. Among these approaches, the Gaussian Mixture Moel (GMM) -based mapping method [8] is widely used, and a number of improvements have been proposed. Toda et al. [9] introduced dynamic features and the global variance (GV) of the converted spectra over a time sequence. Helander et al. [10] proposed transforms based on Partial Least Squares (PLS) to prevent the over-fitting problem encountered in standard multivariate regression.

However, the GMM-based approaches rely on "shallow" voice conversion, a method based on piecewise linear transformation. The shape of the vocal tract is generally non-linear, so non-linear voice conversion is more compatible with human speech. To capture the characteristics of speech more precisely, it is necessary to have a deeper non-linear architecture with more hidden layers. One example of deeper VC methods is proposed by Desai *et al.* [11] based on Neural Networks (NN). Nakashika *et al.* [12] also proposed a VC method using speaker-dependent restricted Boltzmann machines (RBMs) or deep belief networks (DBNs [13]) to achieve non-linear deep transformation. Wu *et al.* [14] utilized a conditional restricted Boltzmann machine (CRBM [15]) to obtain latent non-linear relationships between the speech of a source and that of a target speaker. It was reported that these non-linear VC approaches achieved relatively higher performance than linear transformation approaches [11, 12, 14].

In this paper, we extend our earlier work in [12] to systematically capture time information as well as latent (deep) relationships between a source speaker's and a target speaker's features in a single network, accomplished by combining speaker-dependent CRBMs and a concatenating NN. CRBM is a non-linear probabilistic model used to represent time series data that consists of three factors: (i) an undirected model between binary latent variables and the current visible variables, (ii) a directed model from the previous visible variables to the current visible variables, and (iii) a directed model from the pre-visible variables to the latent variables. In our approach, we first train two exclusive CRBMs for the source and the target speakers independently using segmented training data prepared for each speaker, then train a NN using the projected features, and finally fine-tune the networks as a single network for VC. Because the training data for the source speaker CRBM include various phonemes particular to the speaker, the speaker-dependent network tries to capture the abstractions to maximally express the training data that have abundant speaker individuality information and less phonological information. Furthermore, the network inputs a collection of time-series feature vectors (e.g. the directed models (ii) and (iii) absorb time-related information), so the latent space captures the remaining information (the time-invariant features). Therefore, we expect that if feature conversion is conducted in such time-invariant, individuality-emphasized high-order spaces, it is much easier to convert voice features than if using the original spectrum-based space.

## 2. PRELIMINARIES

Our voice conversion system uses conditional restricted Boltzmann machines (CRBMs) to capture high-order conversion-friendly features. We briefly review the CRBM and its fundamental model, the restricted Boltzmann machine (RBM), in this section.

### 2.1. RBM

RBM is an undirected graphical model that defines the distribution of visible variables with binary hidden (latent) variables [16]. The joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ of binary-valued visible units $\boldsymbol{v} = [v_1, \cdots, v_I]^T, v_i \in \{0, 1\}$ and binary-valued hidden units

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

$\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0, 1\}$ is defined as follows:

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

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^T \boldsymbol{v} - \boldsymbol{c}^T \boldsymbol{h} - \boldsymbol{v}^T \boldsymbol{W} \boldsymbol{h} \tag{2}$$

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

where, $\boldsymbol{W} \in \mathbb{R}^{I \times J}$, $\boldsymbol{b} \in \mathbb{R}^{I \times 1}$, and $\boldsymbol{c} \in \mathbb{R}^{J \times 1}$ are the weight-parameter matrix between visible units and hidden units, a bias vector of visible units, and a bias vector of hidden units, respectively.

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:

$$p(h_j = 1|\boldsymbol{v}) = \sigma(c_j + \boldsymbol{v}^T \boldsymbol{W}_{:j}) \tag{4}$$

$$p(v_i = 1|\boldsymbol{h}) = \sigma(b_i + \boldsymbol{h}^T \boldsymbol{W}_{i:}^T) \tag{5}$$

where $\boldsymbol{W}_{:j}$ and $\boldsymbol{W}_{i:}$ denote the j-th column vector and the i-th row vector, respectively, and $\sigma(x)$ indicates an element-wise sigmoid function; i.e., $\sigma(x) = 1./(1 + e^{-x})$.

For parameter estimation, the log-likelihood of a collection of visible units $\mathcal{L} = \log \prod_n p(\boldsymbol{v}_n)$ is used as an evaluation function. Differentiating partially with respect to each parameter, we obtain:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model} \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial c_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model} \tag{8}$$

where, $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{model}$ indicate expectations of input data and the inner model, respectively. However, it is generally difficult to compute the second term, so typically, expectation of the reconstructed data $\langle \cdot \rangle_{recon}$ computed by Eqs. (4) and (5) is alternatively used [13]. Using Eqs. (6), (7), and (8), each parameter can be updated by stochastic gradient descent.

### 2.2. CRBM

CRBM is an extended version of RBM proposed by Taylor *et al.* [15], and is suitable for the representation of time series data. In addition to the use of an undirected model as in RBM, CRBM also employs directed models between binary hidden units $\boldsymbol{h}^{(t)} = [h_1^{(t)}, \cdots, h_J^{(t)}]^T, h_j^{(t)} \in \{0, 1\}$ and a collection of binary visible units $\{\boldsymbol{v}^{(p)}\}_{p=t-P}^{t}, \boldsymbol{v}^{(p)} = [v_1^{(p)}, \cdots, v_I^{(p)}]^T, v_i^{(p)} \in \{0, 1\}$ at the current frame $t$. For simplicity, we choose $P = 1$ in this paper ($P$ is the number of previous frames from the current frame taken into account). In this model, there are three types of parameters to be estimated: $\boldsymbol{W}_{v'v} \in \mathbb{R}^{I \times I}$ (a directed weight matrix from $\boldsymbol{v}^{(t-1)}$ to $\boldsymbol{v}^{(t)}$), $\boldsymbol{W}_{v'h} \in \mathbb{R}^{I \times J}$ (a directed weight matrix from $\boldsymbol{v}^{(t-1)}$ to $\boldsymbol{h}^{(t)}$), and $\boldsymbol{W}_{vh} \in \mathbb{R}^{I \times J}$ (an undirected weight matrix between $\boldsymbol{v}^{(t)}$ and $\boldsymbol{h}^{(t)}$). These weights are estimated using contrastive divergence in a similar manner to RBM by minimizing the following likelihood:

$$p(\boldsymbol{v}^{(t)}|\boldsymbol{v}^{(t-1)}) = \frac{1}{Z} \sum_{\boldsymbol{h}^{(t)}} e^{-E(\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)}|\boldsymbol{v}^{(t-1)})} \tag{9}$$

where $Z$ is a normalized term, and the energy function $E$ becomes:

$$\begin{aligned} E(\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)}|\boldsymbol{v}^{(t-1)}) = -\boldsymbol{b}^T \boldsymbol{v}^{(t)} - \boldsymbol{c}^T \boldsymbol{h}^{(t)} - {\boldsymbol{v}^{(t)}}^T \boldsymbol{W}_{vh} \boldsymbol{h}^{(t)} \\ - {\boldsymbol{v}^{(t-1)}}^T \boldsymbol{W}_{v'v} \boldsymbol{v}^{(t)} - {\boldsymbol{v}^{(t-1)}}^T \boldsymbol{W}_{v'h} \boldsymbol{h}^{(t)}. \end{aligned} \tag{10}$$

We obtain the following partial differential equations to the log-likelihood $\mathcal{L} = \log \prod_t p(\boldsymbol{v}^{(t)}|\boldsymbol{v}^{(t-1)})$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{v'v_{i'i}}} = \langle v_i^{(t)} v_{i'}^{(t-1)} \rangle_{data} - \langle v_i^{(t)} v_{i'}^{(t-1)} \rangle_{model} \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{v'h_{i'j}}} = \langle v_{i'}^{(t-1)} h_j^{(t)} \rangle_{data} - \langle v_{i'}^{(t-1)} h_j^{(t)} \rangle_{model} \tag{12}$$

The other parameters related to the undirected model ($\boldsymbol{W}_{vh}$, $\boldsymbol{b}$ and $\boldsymbol{c}$) are also calculated from equations (6), (7) and (8) by proper substitution of variables. Once the parameters are estimated, forward inference (the conditional probability of $\boldsymbol{h}^{(t)}$ given $\boldsymbol{v}^{(t)}$ and $\boldsymbol{v}^{(t-1)}$) and backward inference (the conditional probability of $\boldsymbol{v}^{(t)}$ given $\boldsymbol{h}^{(t)}$ and $\boldsymbol{v}^{(t-1)}$) are respectively written as:

$$p(h_j^{(t)} = 1|\boldsymbol{v}^{(t)}, \boldsymbol{v}^{(t-1)}) = \sigma(c_j + {\boldsymbol{v}^{(t)}}^T \boldsymbol{W}_{vh_{:j}} + {\boldsymbol{v}^{(t-1)}}^T \boldsymbol{W}_{v'h_{:j}}) \tag{13}$$

$$p(v_i^{(t)} = 1|\boldsymbol{h}^{(t)}, \boldsymbol{v}^{(t-1)}) = \sigma(b_i + {\boldsymbol{h}^{(t)}}^T \boldsymbol{W}_{vh_{i:}}^T + {\boldsymbol{v}^{(t-1)}}^T \boldsymbol{W}_{v'v_{:j}}). \tag{14}$$

## 3. PROPOSED VOICE CONVERSION

In general, the less phonological and the more individuality-emphasized features a source input includes for a speaker, the easier it is to convert the source features to target features. This paper proposes voice conversion using such features.

Fig. 1 shows an overview of our proposed voice conversion system. In our approach, we independently train CRBMs for each speaker beforehand as shown in Fig. 1 (a). Parameters $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$ ($\boldsymbol{x}^{(t-1)}$ and $\boldsymbol{y}^{(t-1)}$) are acoustic feature vectors (e.g. visible units in CRBM), such as MFCC, at frame $t$ (at frame $t-1$) for a source speaker (and a target speaker). The CRBM described in subsection 2.2 feeds binary-valued visible units, so training data for each CRBM are converted to binary using a sigmoid function beforehand.

For the source speaker, for instance, the parameter matrix $\boldsymbol{W}_{xh}$ is estimated so as to maximize the probability of $T$ chained training samples $p(\boldsymbol{x}) = \prod_{t=1}^{T} p(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)})$ where $\boldsymbol{x}^{(0)} = \boldsymbol{0} \in \mathbb{R}^{I \times 1}$. Because each unit in the hidden vector $\boldsymbol{h}_x^{(t)}$ is independent from the others, it captures the *common* characteristics in the visible units. The training data usually include various phonemes and unvarying speaker-specific features; thus, we expect that the extracted features in $\boldsymbol{h}_x^{(t)}$ represent speaker-individual information. Furthermore, since we estimate the time-related matrices $\boldsymbol{W}_{x'h}$, $\boldsymbol{W}_{x'x}$ jointly with the static term $\boldsymbol{W}_{xh}$ as shown in Eq. (10) using the training data, they *absorb* time-related information and $\boldsymbol{W}_{xh}$ can focus on capturing other information. This means that the obtained features in the hidden units $\boldsymbol{h}_x^{(t)}$ also help to capture speaker-individualities that are not related to time. This is true for either the source or the target speaker.

In our approach, we convert such individuality-emphasized features (from $\boldsymbol{h}_x^{(t)}$ to $\boldsymbol{h}_y^{(t)}$) using a neural network (NN) that has $L + 2$ layers ($L$ is the number of hidden layers; typically, $L$ is 0 or 1) as shown in Fig. 1 (b). To train the NN, we use the parallel training set $\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=0}^{T'}$ where $T'$ is the number of frames of the parallel data[1]. During the training stage of the NN, the projected vectors of the source speaker's acoustic features $\boldsymbol{h}_x^{(t)}$ are the inputs, and the projected vectors of the corresponding target speaker's features $\boldsymbol{h}_y^{(t)}$ are outputs, calculated as[2]:

$$\boldsymbol{h}_x^{(t)} = \sigma(\boldsymbol{W}_{xh}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{x'h}\boldsymbol{x}^{(t-1)} + \boldsymbol{c}_x) \tag{15}$$

$$\boldsymbol{h}_y^{(t)} = \sigma(\boldsymbol{W}_{yh}\boldsymbol{y}^{(t)} + \boldsymbol{W}_{y'h}\boldsymbol{y}^{(t-1)} + \boldsymbol{c}_y) \tag{16}$$

from Eqs. (13) and (14), where $\boldsymbol{c}_x$ and $\boldsymbol{c}_y$ are bias vectors of forward inference for a source speaker and a target speaker, respectively. Weight parameters of the NN $\{\boldsymbol{W}_l, \boldsymbol{d}_l\}_{l=0}^{L}$ are estimated to minimize the error between the output $\eta(\boldsymbol{h}_x^{(t)})$ and the target vector $\boldsymbol{h}_y^{(t)}$ as is typical for a NN. Once the weight parameters are estimated, an input vector $\boldsymbol{h}_x^{(t)}$ is converted to:

$$\eta(\boldsymbol{h}_x^{(t)}) = \bigodot_{l=0}^{L} \eta_l(\boldsymbol{h}_x^{(t)}) \tag{17}$$

$$\eta_l(\boldsymbol{h}_x^{(t)}) = \sigma(\boldsymbol{W}_l\boldsymbol{h}_x^{(t)} + \boldsymbol{d}_l) \tag{18}$$

where $\bigodot_{l=0}^{L}$ denotes the composition of $L + 1$ functions. For instance, $\bigodot_{l=0}^{1} \eta_l(\boldsymbol{z}) = \sigma(\boldsymbol{W}_1\sigma(\boldsymbol{W}_0\boldsymbol{z} + \boldsymbol{d}_0) + \boldsymbol{d}_1)$ for a NN with one hidden layer.

To convert the output of the NN to the acoustic features of the target speaker, we just use backward inference of a CRBM using Eq. (14), resulting in:

$$p(\boldsymbol{y}^{(t)}|\boldsymbol{h}_y^{(t)}, \boldsymbol{y}^{(t-1)}) = \sigma(\boldsymbol{W}_{yh}^T\boldsymbol{h}_y^{(t)} + \boldsymbol{W}_{y'y}\boldsymbol{y}^{(t-1)} + \boldsymbol{b}_y) \tag{19}$$

where $\boldsymbol{b}_y$ is a bias vector of backward inference for the target speaker.

Summarizing the above discussion, a voice conversion function of our method from a source acoustic vector $\boldsymbol{x}^{(t)}$ to a target vector $\boldsymbol{y}^{(t)}$ at frame $t$, given the previous vectors $\boldsymbol{x}^{(t-1)}$ and $\boldsymbol{y}^{(t-1)}$, is written as:

$$\boldsymbol{y}^{(t)} = \bigodot_{k=0}^{L+2} \sigma(\boldsymbol{W}_{(k)}\boldsymbol{x}^{(t)} + \boldsymbol{a}_{(k)}(\boldsymbol{x}^{(t-1)}, \boldsymbol{y}^{(t-1)})) \tag{20}$$

where $\boldsymbol{W}_{(k)}$ and $\boldsymbol{a}_{(k)}(\boldsymbol{x}^{(t-1)}, \boldsymbol{y}^{(t-1)})$ denote elements of a set of dynamic parameters $\Theta^{(t)} = \{\boldsymbol{W}, \boldsymbol{a}^{(t)}\}$:

$$\begin{aligned} \boldsymbol{W} &= \{\boldsymbol{W}_{(k)}\}_{k=0}^{L+2} && (21)\\ &= \{\boldsymbol{W}_{xh}, \boldsymbol{W}_0, \cdots, \boldsymbol{W}_L, {\boldsymbol{W}_{yh}}^T\} && (22)\\ \boldsymbol{a}^{(t)} &= \{\boldsymbol{a}_{(k)}(\boldsymbol{x}^{(t-1)}, \boldsymbol{y}^{(t-1)})\}_{k=0}^{L+2} && (23)\\ &= \{\boldsymbol{W}_{x'h}\boldsymbol{x}^{(t-1)} + \boldsymbol{c}_x, \boldsymbol{d}_0, \cdots, \boldsymbol{d}_L, \boldsymbol{W}_{y'y}\boldsymbol{y}^{(t-1)} + \boldsymbol{b}_y\}. && (24) \end{aligned}$$

The conversion function shown in Eq. (20) implies a dynamic model of a $(L + 4)$-layer NN with sigmoid activated functions. Therefore, we can fine-tune each parameter of the entire network consisting of the two CRBMs and the NN by back-propagation using the acoustic parallel data.

As Eq. (20) indicates, we need a current acoustic vector from a source speaker, and previous vectors from both a source speaker and a target speaker to estimate the target speaker's current acoustic vector. However, we never know the correct previous vector of the target speaker, so in practice, we use the last converted (estimated) vectors as the previous target vector iteratively, starting from a zero vector. We confirmed that this approach worked well through our preliminary experiments.

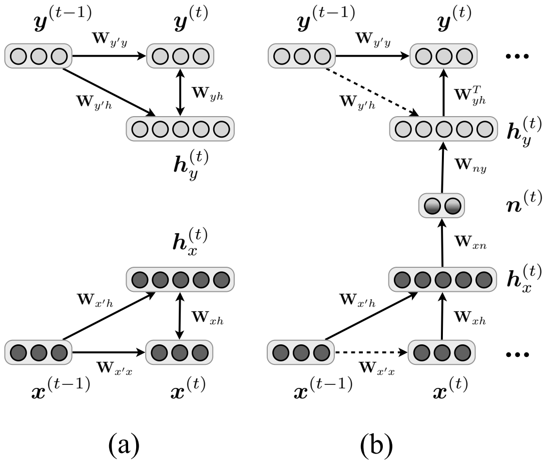

**Fig. 1**. (a) CRBMs for a source speaker (below) and a target speaker (above), (b) our proposed voice conversion architecture combining two speaker-dependent CRBMs with a NN.

Meanwhile, a conventional GMM-based approach [9] with $M$ Gaussian mixtures converts the source features $\boldsymbol{x}$ as

$$\boldsymbol{y} = \sum_{m=1}^{M} P(m|\boldsymbol{x})(\boldsymbol{\Sigma}_{yx}^{(m)}\boldsymbol{\Sigma}_{xx}^{(m)-1}(\boldsymbol{x} - \boldsymbol{\mu}_x^{(m)}) + \boldsymbol{\mu}_y^{(m)}) \tag{25}$$

$$P(m|\boldsymbol{x}) = \frac{w^{(m)}\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x^{(m)}, \boldsymbol{\Sigma}_{xx}^{(m)})}{\sum_{m=1}^{M} w^{(m)}\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x^{(m)}, \boldsymbol{\Sigma}_{xx}^{(m)})} \tag{26}$$

and is an additive model of non-linear functions. Our approach using Eq. (20) is based on the composite function of multiple different non-linear functions feeding time-series data. Therefore, it is expected that our composite model can represent more complex relationships than the conventional GMM-based method and other static network approaches [11, 12].

## 4. EXPERIMENTS

### 4.1. Setup

We conducted voice conversion experiments using the ATR Japanese speech database [17], comparing our method ("Our") with the well-known GMM-based approach ("GMM"), conventional NN-based voice conversion ("NN") and our previous work [12] ("DBN"). From this database, we used a male speaker (MMY) for the source, and a female speaker (FTK) for the target. As an input vector, 24-dimensional MFCC features were calculated from STRAIGHT spectra [18] using filter-theory [19] to decode the MFCC back to STRAIGHT spectra in the synthesis stage. For the GMM-based approach (64 mixtures), we further calculated delta (24 dimensions) and delta-delta (24 dimensions) features as typically done for GMM-based VC, and concatenated them as a super vector (72 dimensions in total) to provide a fair comparison with our method (that operates using multiple frames $\boldsymbol{x}^{(t)}$ and $\boldsymbol{x}^{(t-1)}$). The parallel data of the

[1] For sake of simplicity, we used the same parallel data for both training of the CRBMs and the NN in our experiments ($T' = T$).

[2] We use the expected values of $\boldsymbol{h}_x^{(t)}$ and $\boldsymbol{h}_y^{(t)}$ as the latent features.

**Table 1**. Various architectures used for the preliminary experiment.

| Arcitectures | NN | DBN [12] | Our method | Layers |
|---|---|---|---|---|
| arc. 1 | [24-24-24-24] | [24:24-24:24] | [(24,24):24-(24,24):24] | 4 |
| arc. 2 | [24-48-48-24] | [24:48-48:24] | [(24,24):48-(48,24):24] | 4 |
| arc. 3 | [24-24-24-24-24-24] | [24:24:24-24:24:24] | [(24,24):24:24-24:(24,24):24] | 6 |
| arc. 4 | [24-48-24-24-48-24] | [24:48:24-24:48:24] | [(24,24):48:24-24:(48,24):24] | 6 |

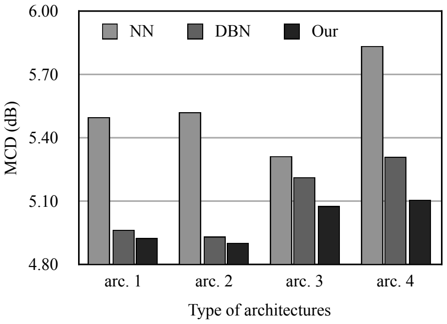

**Fig. 2**. Averaged mel-cepstral distortion with changing network architetures ($N = 10,000$).

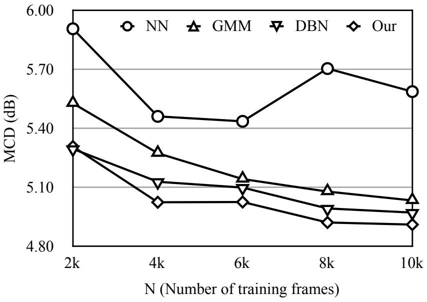

**Fig. 3**. Averaged mel-cepstral distortion for each method.

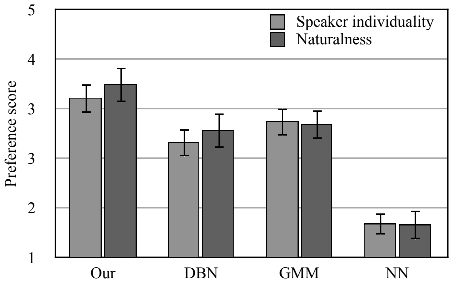

**Fig. 4**. MOS scores w.r.t. speaker individuality and naturalness. The error bars show 95% confidence intervals.

source/target speakers processed by Dynamic Programming were created from 216 word utterances in the dataset, and were used for training. Note that the parallel data were prepared for the NN and GMM methods, and two speaker-wise CRBMs were trained independently. We set the learning rate and the number of epochs in the gradient descent-based training of CRBMs be 0.01 and 200, respectively. For the objective test, 20 sentences (about 70 sec. long) that were not included in the training data were arbitrarily selected from the database. For the objective evaluation, we used MCD (mel-cepstral distortion) to measure how close the converted vector is to the target vector in mel-cepstral space. We calculated the MCD for each frame in the training data, and averaged the MCD values for the final evaluation.

### 4.2. Evaluation

We first investigated how our approach works as the architecture of the VC network changes, comparing it to conventional NN-based VC and DBN-based VC with similar architecture. In this preliminary experiment, 4 types of architectures are compared, where we changed the number of layers and the number of units in each layer as listed in Table 1. The values in the table indicate the number of units from the source layer to the target layer. For our method, for instance, the numbers are described as *[CRBM for source - NN - CRBM for target]*. Fig. 2 compares the averaged MCD obtained for each architecture. As shown in Fig. 2, the deeper architecture (such as “arc. 3”) does not always provide better results than shallower architectures. The best architecture using our method was “arc. 2”, so the four-layer architecture “arc. 2” is used for all the remaining experiments reported in this paper.

Fig. 3 and Fig. 4 summarize the experimental results, comparing each method with respect to objective and subjective criteria, respectively. For the subjective evaluation, MOS (mean opinion score) listening tests were conducted, where 9 participants listened to pairs of an original target speech signal (generated from analysis-by-synthesis) and the converted speech signals for each method, and then selected how close the converted speech sounded to the original one in terms of speaker individuality and naturalness on a 5-point scale (5: excellent, 4: good, 3: fair, 2: poor, and 1: bad). As shown in these figures, our approach (“Our”) outperformed other conventional methods (GMM, DBN and NN) in both criteria (all differences are significant at a significance level of 0.01). The reason for the improvement is attributed to the fact that our time-invariant high-order conversion system using CRBMs is able to capture and convert the abstractions of speaker individualities better than the other methods. Especially as shown in Fig. 4, our approach achieved high performance in naturalness. This is because the CRBMs captured time-related information and alleviated jaggy noise caused by frame-by-frame conversion.

## 5. CONCLUSION

We presented a voice conversion method that combines speaker-dependent CRBMs and a NN to extract speaker-individual information for specch conversion. Through experiments, we showed improvement of speech conversion in terms of MCD and MOS criteria when compared with the well-known conventional GMM-based approach and other network-based approaches.

## 6. REFERENCES

[1] Alexander Kain and Michael W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998, pp. 285–288.

[2] Christophe Veaux and X. Robet, "Intonation conversion from neutral to expressive speech," in *Proc. 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 *IEEE International Conference on Acoustics, Speech, and Signal Processing (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 *Proc. Interspeech*, 2009, pp. 308–311.

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

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

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

[9] Tomoki Toda, Alan W. Black, and Keiichi 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.

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

[11] Srinivas Desai, E. Veera Raghavendra, B. Yegnanarayana, Alan W. Black, and Kishore Prahallad, "Voice conversion using artificial neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 3893–3896.

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

[13] 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.

[14] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Conditional restricted boltzmann machine for voice conversion," in *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, 2013.

[15] 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.

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

[17] 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.

[18] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 3933–3936.

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