Voice Conversion Using RNN Pre-Trained by Recurrent Temporal Restricted Boltzmann Machines

Toru Nakashika, *Member, IEEE*, Tetsuya Takiguchi, *Member, IEEE*, and Yasuo Ariki, *Member, IEEE*

***Abstract*—This paper presents a voice conversion (VC) method that utilizes the recently proposed probabilistic models called recurrent temporal restricted Boltzmann machines (RTRBMs). One RTRBM is used for each speaker, with the goal of capturing high-order temporal dependencies in an acoustic sequence. Our algorithm starts from the separate training of one RTRBM for a source speaker and another for a target speaker using speaker-dependent training data. Because each RTRBM attempts to discover abstractions to maximally express the training data at each time step, as well as the temporal dependencies in the training data, we expect that the models represent the linguistic-related latent features in high-order spaces. In our approach, we convert (match) features of emphasis for the source speaker to those of the target speaker using a neural network (NN), so that the entire network (consisting of the two RTRBMs and the NN) acts as a deep recurrent NN and can be fine-tuned. Using VC experiments, we confirm the high performance of our method, especially in terms of objective criteria, relative to conventional VC methods such as approaches based on Gaussian mixture models and on NNs.**

***Index Terms*—Deep Learning, recurrent neural network, recurrent temporal restricted Boltzmann machine (RTRBM), speaker specific features, voice conversion.**

I. Introduction

VOICE conversion (VC) techniques, by which specific information in the speech of a source speaker is transformed into that of a target speaker while retaining linguistic information, have recently attracted 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]. In this report, as in most VC work, we focus not on fundamental frequency (f0) conversion, but on the conversion of spectral features.

Various statistical approaches to VC have been studied, including those discussed in [6], [7]. Among these approaches, mapping methods based on the Gaussian mixture model (GMM) [8] are widely used, and a number of improvements have been proposed. Some of the approaches do not require parallel data, because they use a GMM-adaptation technique [9], eigen-voice GMM [10], [11] or probabilistic integration model [12].

Manuscript received May 16, 2014; revised September 15, 2014; accepted November 23, 2014. Date of publication December 09, 2014; date of current version February 26, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhen-Hua Ling.

T. Nakashika is with the Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan (e-mail: nakashika@me.cs.scitec.kobe-u.ac.jp).

T. Takiguchi and Y. Ariki are with Organization of Advanced Science and Technology, Kobe University, Kobe 657-8501, Japan.

Digital Object Identifier 10.1109/TASLP.2014.2379589

However, GMM-based approaches rely on "shallow" voice conversion—methods are based on piecewise-linear transformations. Because the shape of the vocal tract is generally non-linear, non-linear voice conversion is more compatible with human speech. To capture the characteristics of speech more precisely, a deeper non-linear architecture with more hidden layers is required. One example of deeper VC methods was proposed by Desai *et al.* [13] based on neural networks (NN). In GMM-based approaches, the conversion is achieved so as to maximize the conditional probability calculated from a joint probability of source and target speech, where the joint model is trained beforehand. In contrast, NN-based approaches directly train the conditional probability, which converts the feature vector of a source speaker to that of a target speaker. These approaches have been shown to perform better than generative approaches, such as GMM-based approaches, in speech recognition and synthesis, as well as in VC [14], [15]. For these reasons, NN-based approaches achieve relatively high performance if the training samples are carefully prepared [13].

These approaches often suffer from over-smoothing or over-fitting problems. GMM-based approaches represent acoustic features using multiple Gaussian distributions, which are estimated by averaging observations with similar context descriptions in the training. Therefore, the outputs of a GMM distribute near the modes (means) of the Gaussians, which leads to problems with over-smoothing. Furthermore, over-fitting problems arise when we give more Gaussian mixtures due to precise estimation of the observed distribution. In NN-based approaches, the model is often over-fitted due to its complexity. The model exaggerates small fluctuations in the unknown data if the amount of training data is not sufficient for the number of parameters.

Some methods have been proposed for alleviating the over-smoothing of GMMs, such as the global variance model [16], a minimizing-divergence model [17], and post-filtering [18]. Other approaches that reduce over-smoothing have also been proposed, such as canonical correlation analysis (CCA) [19] and an exemplar-based VC system using NMF (non-negative matrix factorization) [20], [21]. In our earlier work [22], we proposed a new VC technique that copes with over-fitting problems in NN-based approaches using a combination of speaker-dependent restricted Boltzmann machines (RBMs) [23] (or deep belief nets; DBN [24]), which capture high-order features in an unsupervised manner, and concatenating NNs. These graphical models are better than GMMs at representing the distribution of high-dimensional observations with cross-dimensional correlations in speech synthesis [25] and in speech recognition [26]. Since Hinton *et al.* introduced an effective training algorithm

for the DBN in 2006 [24], the use of deep learning has rapidly spread in the field of signal processing, as well as in speech signal processing. RBMs (or DBNs) have been used, for example, for recognition of handwritten characters [24], recognition of 3-D objects [27], and machine transliteration [28].

In this paper, we extend our earlier work in [22] to systematically capture time information as well as latent (deep) relationships between source-speaker and target-speaker features in a single network. We do this by combining speaker-dependent recurrent temporal restricted Boltzmann machines (RTRBMs [29]) and a concatenating NN. An RTRBM, which is an extension of an RBM, is a non-linear probabilistic model used to capture temporal dependencies in time-series data. Despite its simplicity, this model does a good job of describing meaningful sequences such as video [29] and music [30]. In our approach, we first train two RTRBMs: one exclusively for the source and one exclusively for the target speakers. We train them independently using segmented training data prepared for each speaker. Then we train an NN using the projected features. RBM families, including RTRBM, try to capture latent features that appear sparsely in relation to each other due to their restriction. Therefore, when we train an RBM or an RTRBM using training data that includes various phonemes, the system tries to suppress speaker-specific information and capture linguistical or phonological-related information, since the phonemes appear exclusively at each frame. Furthermore, the RTRBM discovers temporal correlations in the *phonological* space, unlike the traditional RBMs. This suggests that it helps to capture relationships between phonemes. Each speaker-dependent RTRBM can encode acoustic features to such phonological features that suppress the speaker's specificity in a forward inference stage, and also decode the phonological features back to the acoustic features in a backward inference stage. The concatenating NN possibly acts to match indexes of the source speaker's latent features to those of the target speaker's. Consequently, our VC method, which includes the encoding of the acoustic features using the source speaker's RTRBM, matching of the feature indexes using the NN, and decoding to the acoustic features using the target speaker's RTRBM, is based on deep non-linear transformation. Furthermore, the whole network can also be fine-tuned, regarding it as a single recurrent NN.

Similar research can be found in [31] and [32]. Wu *et al.* employed a conditional restricted Boltzmann machine (CRBM), another model for representing time-series data that was proposed by Taylor *et al.* [33] for capturing linear and non-linear relationships between source and target features [31]. Chen *et al.* also used an RBM to model the joint spectral distribution instead of using a conventional joint-density GMM [32]. Unlike these approaches, our method has the ability to model temporal characteristics in acoustic feature sequence and to model deeper model structure, making it possible to model higher non-linear conversion relationships.

The rest of the article is organized as follows. In Section II, we briefly review the fundamental techniques, (RBMs and RTRBMs) before explaining our method. The proposed VC system is presented in Section III. We describe the various experiments and VC results in Section IV, and we conclude the article in Section V.

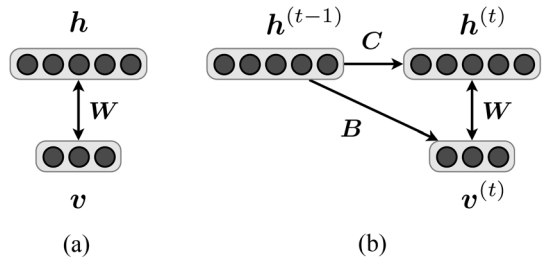

Fig. 1. Graphical representation of (a) an RBM and (b) an RTRBM.

II. Probabilistic Models

Our voice conversion system uses RTRBMs to capture high-order conversion-friendly features. In this section we briefly review RBMs as a fundamental model, and the RTRBMs.

A. RBM

RBMs [34] were originally introduced as undirected graphical models for defining the distribution of binary visible variables with binary hidden (latent) variables, as shown in Fig. 1(a). Later, this model was extended to deal with data with real values in the Gaussian-Bernoulli RBM (GBRBM) [24], and became a popular tool for representing complicated distributions of actual data, such as audio and images. In the literature of an improved GBRBM [35], 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 $\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0, 1\}$ is defined with an energy function $E(\boldsymbol{v}, \boldsymbol{h})$ as follows:

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

$$E(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{2} \left\| \frac{\boldsymbol{v} - \boldsymbol{b}}{\boldsymbol{\sigma}} \right\|^2 - \boldsymbol{c}^T \boldsymbol{h} - \left(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2} \right)^T \boldsymbol{W} \boldsymbol{h} \tag{2}$$

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

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 parameters for 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. (2) denotes the element-wise division.

Because neither visible nor hidden units are connected to each other, the conditional probabilities $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ form simple equations as follows:

$$p(h_j = 1|\boldsymbol{v}) = \mathcal{S}(c_j + \boldsymbol{W}_{:j}^T (\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2})) \tag{4}$$

$$p(v_i = v|\boldsymbol{h}) = \mathcal{N}(v|b_i + \boldsymbol{W}_{i:} \boldsymbol{h}, \sigma_i^2), \tag{5}$$

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 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

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ij}} = \langle \frac{v_i h_j}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i h_j}{\sigma_i^2} \rangle_{model} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \langle \frac{v_i}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i}{\sigma_i^2} \rangle_{model} \quad (7)$$

$$\frac{\partial \mathcal{L}}{\partial c_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}, \quad (8)$$

where $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{model}$ indicate the expectations of the input data and the inner model, respectively. However, it is generally difficult to compute the second term. Therefore, the expectation of the reconstructed data $\langle \cdot \rangle_{recon}$ computed by Eqs. (4) and (5) is used instead [24]. Based on Eqs. (6), (7), and (8), each parameter can be updated using stochastic gradient descent.

B. RTRBM

An RTRBM is an extended RBM proposed by Sutskever *et al.* [29], and is suitable for capturing and modeling temporal dependencies in sequence data. Like an RBM, an RTRBM uses an undirected model. An RTRBM also employs directed models from previous hidden units $\boldsymbol{h}^{(t-1)} = [h_1^{(t-1)}, \cdots, h_J^{(t-1)}]^T$, $h_j^{(t-1)} \in \{0, 1\}$ to current hidden units $\boldsymbol{h}^{(t)} = [h_1^{(t)}, \cdots, h_J^{(t)}]^T$, $h_j^{(t)} \in \{0, 1\}$ and current visible units $\boldsymbol{v}^{(t)} = [v_1^{(t)}, \cdots, v_I^{(t)}]^T$, $v_i^{(t)} \in \mathbb{R}$ at the current frame t as shown in Fig. 1(b). In this model, there are three types of parameters to be estimated: $\boldsymbol{B} \in \mathbb{R}^{I \times J}$ (a directed weight matrix from $\boldsymbol{h}^{(t-1)}$ to $\boldsymbol{v}^{(t)}$), $\boldsymbol{C} \in \mathbb{R}^{J \times J}$ (a directed weight matrix from $\boldsymbol{h}^{(t-1)}$ to $\boldsymbol{h}^{(t)}$), and $\boldsymbol{W} \in \mathbb{R}^{I \times J}$ (an undirected weight matrix between $\boldsymbol{v}^{(t)}$ and $\boldsymbol{h}^{(t)}$). Like with an RBM, the weights are estimated using contrastive divergence by maximizing the log-likelihood $\mathcal{L} = \log \prod_t p(\boldsymbol{v}^{(t)} | \mathcal{A}^{(t)})$ denoted by $\mathcal{A}^{(t)} = \{\boldsymbol{v}^{(\tau)}, \boldsymbol{h}^{(\tau)} | \tau < t\}$, where

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

In our RTRBM model, the energy function E becomes

$$E(\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)} | \boldsymbol{h}^{(t-1)}) = \frac{1}{2} \left\| \frac{\boldsymbol{v}^{(t)} - \boldsymbol{b}^{(t)}}{\boldsymbol{\sigma}} \right\|^2 - {\boldsymbol{c}^{(t)}}^T \boldsymbol{h}^{(t)} - \left(\frac{\boldsymbol{v}^{(t)}}{\boldsymbol{\sigma}^2} \right)^T \boldsymbol{W} \boldsymbol{h}^{(t)} \quad (10)$$

$$\boldsymbol{b}^{(t)} = \boldsymbol{b} + \boldsymbol{B} \boldsymbol{h}^{(t-1)} \quad (11)$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{c} + \boldsymbol{C} \boldsymbol{h}^{(t-1)}. \quad (12)$$

The previous hidden units $\boldsymbol{h}^{(t-1)}$ in Eqs. (11) and (12) are replaced with the mean-field values $\hat{\boldsymbol{h}}^{(t-1)}$ as follows:

$$\hat{\boldsymbol{h}}^{(t-1)} = \mathcal{S}(\boldsymbol{c}^{(t-1)} + \boldsymbol{W}^T (\frac{\boldsymbol{v}^{(t-1)}}{\boldsymbol{\sigma}^2})) \quad (13)$$

since this approach improves the efficiency of training [29]. In this paper, for the initial values $\boldsymbol{h}^{(0)}$, we use a zero vector.

We obtain the following partial differential equations for the log-likelihood:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{B}_{ij}} = \langle \frac{v_i^{(t)} \hat{h}_j^{(t-1)}}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i^{(t)} \hat{h}_j^{(t-1)}}{\sigma_i^2} \rangle_{model} \quad (14)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{C}_{j'j}} = \langle \hat{h}_{j'}^{(t-1)} \hat{h}_j^{(t)} \rangle_{data} - \langle \hat{h}_{j'}^{(t-1)} \hat{h}_j^{(t)} \rangle_{model}. \quad (15)$$

The other parameters related to the undirected model ($\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{c}$) are also calculated from Eqs. (6), (7) and (8) by proper substitution of variables. The second terms in Eqs. (14) and (15) are computed as the reconstructed values in a way similar to that in an RBM.

After the parameters are estimated, forward inference (the conditional probability of $\boldsymbol{h}^{(t)}$ given $\boldsymbol{v}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$) and backward inference (the conditional probability of $\boldsymbol{v}^{(t)}$ given $\boldsymbol{h}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$) can be written as follows, respectively:

$$p(h_j^{(t)} = 1 | \boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t-1)}) = \mathcal{S}(c_j^{(t)} + \boldsymbol{W}_{:j}^T (\frac{\boldsymbol{v}^{(t)}}{\boldsymbol{\sigma}^2})) \quad (16)$$

$$p(v_i^{(t)} = v | \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(t-1)}) = \mathcal{N}(v | b_i^{(t)} + \boldsymbol{W}_{i:} \boldsymbol{h}^{(t)}, \sigma_i^2). \quad (17)$$

III. Voice Conversion Using SD-RTRBMs

In our method, we first try to encode the source speaker's acoustic features to the linguistic information, and decode it into the target speaker's acoustic features. For capturing such linguistic information from the specific speaker's speech, we adopt speaker-dependent recurrent temporal restricted Boltzmann machines (SD-RTRBMs).

Fig. 2 shows an overview of our proposed voice conversion system. Fig. 2(a) shows how we independently train RTRBMs for each speaker beforehand. Variables $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$ ($\boldsymbol{x}^{(t-1)}$ and $\boldsymbol{y}^{(t-1)}$) are acoustic feature vectors (i.e., visible units in RTRBM), such as mel-frequency cepstral coefficients (MFCC), at frame t (at frame $t-1$) for a source speaker and a target speaker, respectively.

For the source speaker, for example, the parameter matrices $\boldsymbol{W}_x$, $\boldsymbol{B}_x$, and $\boldsymbol{C}_x$ are estimated so as to maximize the probability of a T-time sequence $p(\boldsymbol{x}) = \prod_t^T p(\boldsymbol{x}^{(t)} | \mathcal{A}^{(t)})$. Because each unit in the hidden vector $\boldsymbol{h}_x^{(t)}$ is independent from the others, the units capture the *common* characteristics in the visible units. The training data usually include various phonemes; thus, we expect that the extracted features in $\boldsymbol{h}_x^{(t)}$ emphasize linguistic information commonly seen in acoustic vectors. Furthermore, because we estimate the time-related matrices $\boldsymbol{B}_x$ and $\boldsymbol{C}_x$ jointly with the static term $\boldsymbol{W}_x$ as shown in Eq. (10) using the training data, the matrices capture time-related information. This means that the obtained features in the hidden units $\boldsymbol{h}_x^{(t)}$ also help to capture time-related features. An input vector $\boldsymbol{x}^{(t)}$ at frame t is projected into such the latent space that captures linguistic information. In this paper, the latent features $\boldsymbol{h}_x^{(t)}$ are obtained using mean-field approximation as in Eq. (16). The above discussion applies to the target speaker, and the hidden vector for the target $\boldsymbol{h}_y^{(t)}$ is obtained in the same manner. In our approach, we convert (match) such linguistic-information-emphasized features (from $\boldsymbol{h}_x^{(t)}$ to $\boldsymbol{h}_y^{(t)}$) using an NN with $L+2$ layers (L denotes the number of hidden layers; typically, L is 0 or 1), as shown in Fig. 2(b). To train the NN, we use the parallel training set $\{\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}\}_{t=0}^{T'}$

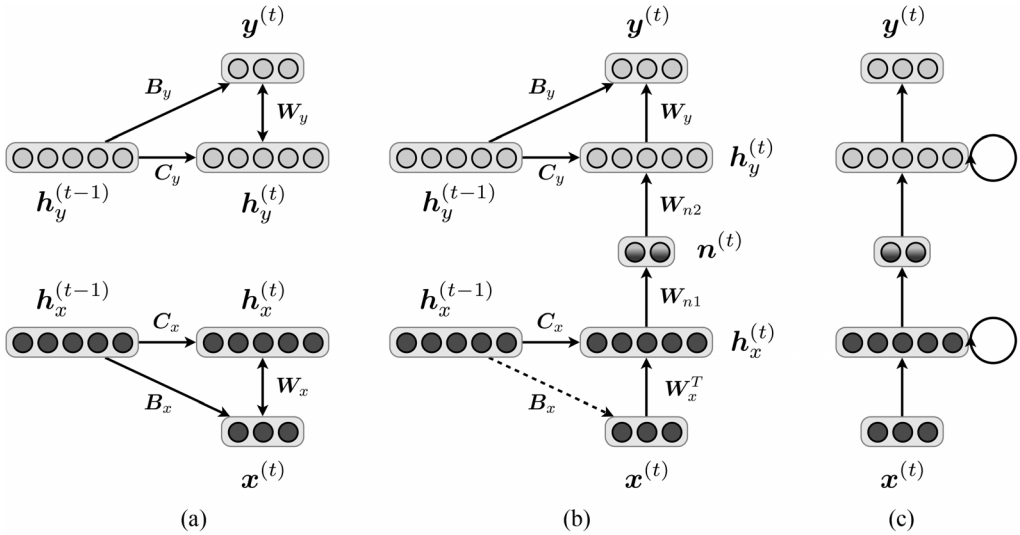

Fig. 2. (a) RTRBMs for a source speaker (below) and a target speaker (above), (b) our proposed voice conversion architecture, which combines two speaker-dependent RTRBMs with an NN, (c) an alternative representation of (b) that can be regarded as a recurrent NN.

where T' denotes 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. 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. After the weight parameters are estimated, an input vector $\boldsymbol{h}_x^{(t)}$ is converted as follows:

$$\eta(\boldsymbol{h}_x^{(t)}) = \bigodot_{l=o}^{L} \eta l(\boldsymbol{h}_x^{(t)}) \tag{18}$$

$$\eta_l(\boldsymbol{h}_x^{(t)}) = \mathcal{S}(\boldsymbol{W}_l \boldsymbol{h}_x^{(t)} + \boldsymbol{d}_l) \tag{19}$$

where $\odot_{l=0}^{L}$ denotes the composition of $L+1$ functions. For example, $\odot_{l=0}^{1} \eta l(Z) = \mathcal{S}(\boldsymbol{W}_1 \mathcal{S}(\boldsymbol{W}_0 z + \boldsymbol{d}_0) + \boldsymbol{d}_1)$ for an NN with one hidden layer. To convert the output of the NN to the acoustic features of the target speaker, we simply use backward inference of an RTRBM using Eq. (17).

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 is written as:

$$\boldsymbol{y}^{(t)} = \arg\max_{\boldsymbol{y}^{(t)}} p(\boldsymbol{y}^{(t)} | \boldsymbol{x}^{(t)}, \boldsymbol{h}_x^{(t-1)}, \boldsymbol{h}_y^{(t-1)}) \tag{20}$$

$$= \boldsymbol{a}_{L+2}^{(t)} + \boldsymbol{W}_{L+2} \bigodot_{k=0}^{L+2} \mathcal{S}(\boldsymbol{a}_k^{(t)} + \boldsymbol{W}_k \boldsymbol{x}^{(t)}) \tag{21}$$

where $\boldsymbol{a}_k^{(t)}$ and $\boldsymbol{W}_k$ denote elements of a set of dynamic parameters $\Theta^{(t)} = \{\boldsymbol{a}^{(t)}, \boldsymbol{W}\}$:

$$\boldsymbol{a}^{(t)} = \{\boldsymbol{a}_k^{(t)}\}_{k=0}^{L+2} = \{\boldsymbol{c}_x^{(t)}, \boldsymbol{d}_0, \cdots, \boldsymbol{d}_L, \boldsymbol{b}_y^{(t)}\} \tag{22}$$

$$\boldsymbol{W} = \{\boldsymbol{W}_k\}_{k=0}^{L+2} = \{\boldsymbol{W}_x^T, \boldsymbol{W}_0, \cdots, \boldsymbol{W}_L, \boldsymbol{W}_y\}, \tag{23}$$

where $\boldsymbol{c}_x^{(t)}$ and $\boldsymbol{b}_y^{(t)}$ denote a forward-inference bias vector in a source speaker's RTRBM and a backward-inference bias vector in the target speaker's RTRBM obtained from Eqs. (12) and (11), respectively. $\boldsymbol{h}_x^{(0)}$ and $\boldsymbol{h}_y^{(0)}$ are zero vectors. The conversion function shown in Eq. (21) implies an $(L+4)$-layer recurrent NN with sigmoid activated functions as shown in Fig. 2(c). Therefore, we can fine-tune each parameter of the network of two RTRBMs and the NN by back-propagation through time (BPTT [36]) using acoustic parallel data. Specifically, each parameter is re-updated so as to minimize the total error ϵ in a gradient-descent-based approach, which is defined as:

$$\epsilon = \sum_{1 \le t \le T} \epsilon^{(t)} = \frac{1}{2} \sum_{1 \le t \le T} (\boldsymbol{y}^{(t)} - \boldsymbol{\nu}^{(t)})^2, \tag{24}$$

where $\boldsymbol{\nu}^{(t)}$ denotes the output of RNN at frame t. The gradient with respect to θ, which is a parameter in the highest recursive hidden layer, for instance, can be written as follows:

$$\frac{\partial \epsilon}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \epsilon^{(t)}}{\partial \theta} \tag{25}$$

$$\frac{\partial \epsilon^{(t)}}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \epsilon^{(t)}}{\partial \boldsymbol{h}_y^{(t)}} \frac{\partial \boldsymbol{h}_y^{(t)}}{\partial \boldsymbol{h}_y^{(k)}} \frac{\partial^+ \boldsymbol{h}_y^{(k)}}{\partial \theta} \right) \tag{26}$$

$$\frac{\partial \boldsymbol{h}_y^{(t)}}{\partial \boldsymbol{h}_y^{(k)}} = \prod_{t \ge i > k} \frac{\partial \boldsymbol{h}_y^{(i)}}{\partial \boldsymbol{h}_y^{(i-1)}} \tag{27}$$

$$= \prod_{t \ge i > k} \boldsymbol{W}_{y'y} (1 - \mathcal{S}(\boldsymbol{h}_y^{(i-1)})), \tag{28}$$

where $\frac{\partial^+ \boldsymbol{h}^{(k)}}{\partial \theta}$ refers to the immediate partial derivative of the hidden units $\boldsymbol{h}^{(k)}$ with respect to θ (i.e., $\boldsymbol{h}^{(k-1)}$ is regarded as a constant with respect to θ).

In contrast, a conventional GMM-based approach on MMSE (minimum mean square error) criterion [37] with M Gaussian mixtures converts the source features $\boldsymbol{x}$ as

$$\boldsymbol{y} = \sum_{m=1}^{M} P(m|\boldsymbol{x}) (\boldsymbol{\Sigma}_{yx}^{(m)} \boldsymbol{\Sigma}_{xx}^{(m)-1} (\boldsymbol{x} - \boldsymbol{\mu}_x^{(m)}) + \boldsymbol{\mu}_y^{(m)}) \tag{29}$$

$$P(m|\boldsymbol{x}) = \frac{w^{(m)} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x^{(m)}, \boldsymbol{\Sigma}_{xx}^{(m)})}{\sum_{m=1}^{M} w^{(m)} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x^{(m)}, \boldsymbol{\Sigma}_{xx}^{(m)})} \tag{30}$$

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

where $w^{(m)}$, $\boldsymbol{\mu}_{\cdot}^{(m)}$ and $\Sigma_{\cdot}^{(m)}$ denote the weight, the corresponding mean vectors, and the corresponding covariance matrices to the speaker of mixture m, respectively. Thus, the GMM-based approach uses an additive model of piecewise linear functions. Our approach using Eq. (21) is based on the composite function of multiple different non-linear functions representing 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 [13], [22] do.

IV. Experiments

A. Conditions

In our VC experiments, we compared our method (SD-RTRBM) with three conventional methods: a well-known GMM-based approach (GMM), an NN-based approach (NN) and our previous work [22], which utilized speaker-dependent RBMs for pre-training of the NN (SD-RBM). In [22], deep architectures using DBNs were reported, but we used a single-layer DBN (i.e., an RBM) for each speaker for comparison with our method. All of the network-based methods (SD-RTRBM, NN, SD-RBM) contained four layers with various numbers of hidden units as discussed in the following section. We trained the network-based methods with a learning rate of 0.01 and momentum of 0.9, with the number of epochs being 400, using acoustic features from the ATR Japanese speech database [38]. The parameters of our method and SD-RBM were fine-tuned after the training of the RTRBMs and RBMs, respectively. In order to evaluate our method under various circumstances, we tested male-to-female (the source and the target speakers are identified with "MMY" and "FTK" in the database, respectively), female-to-female ("FKN" and "FTK"), and male-to-male ("MMY" and "MHT") patterns. 24-dimensional MFCC features were used as an input vector, calculated from STRAIGHT spectra [39] using filter-theory [40] to decode the MFCC back to STRAIGHT spectra in the synthesis stage. Unlike our previous work [22], we processed the obtained MFCC with ZCA (zero component analysis) whitening [41] for all methods (including SD-RBM and GMM), by which we confirmed that it worked better than without whitening, especially for NN. Parallel data from source and target speakers processed by dynamic programming was obtained from 216 word utterances in the dataset. This data was used for training. Note that the parallel data was prepared for the NN and GMM methods, and two speaker-wise RTRBMs were trained independently. For the objective test, 15 sentences that were not included in the training data were arbitrarily selected from the database. For objective evaluation, we used MCD (mel-cepstral distortion) to measure how close the converted vector was to the target vector in the mel-cepstral space. MCD is defined as

$$MCD[dB] = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}\left(\boldsymbol{c}_d - \boldsymbol{c}'_d\right)^2} \tag{31}$$

where $\boldsymbol{c}_d$ and $\boldsymbol{c}'_d$ denote the d-th original target MFCC and the converted MFCC, respectively. The smaller the MCD value, the closer the converted spectra are to the target spectra. We calculated the MCD for each frame in the training data, and averaged them for the final evaluation.

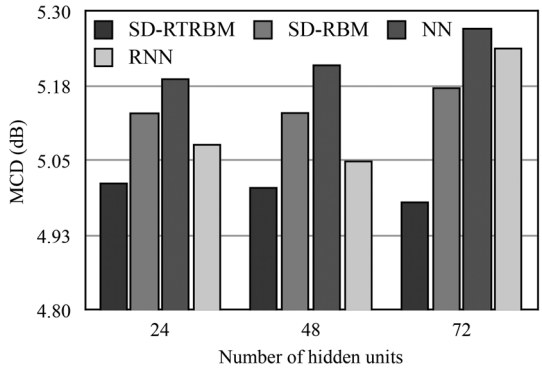

Fig. 3. Average MCD with changing the number of hidden units for each network-based method.

For subjective evaluation, MOS (mean opinion score) listening tests were conducted using the same 15 sentences that were used for objective evaluation. In the subjective evaluation, a male-to-female pair was evaluated. When recovering waveforms from the converted MFCC, we used the correct F0 of the target speaker and the 0th coefficient of the source speaker, just because we want to see the effect of the spectral conversion. For the tests, seven participants listened to the original target speech (generated from analysis-by-synthesis) and converted speech for each method, and then selected how close the converted speech sounded to the original speech on a 5-point scale (5: excellent; 4: good; 3: fair; 2: poor; and 1: bad).

B. Changing the Network Structures

Determining the number of hidden units in the network-based approaches is important for a fair comparison. For reference, we also compared our method with a recurrent neural network (RNN), whose parameters were randomly initialized with the same architecture as our method. All models were trained using $T = 20,000$ frames from the male-to-female training data, and evaluated using a development set of five sentences (identified with SDA16–20 in the database) that were not included in either the training set or the test set.

In the first experiments, we used 24, 48, and 72 hidden units for the network-based approaches and checked the performance of each method. Each network-based method has a four-layer architecture; for example, the 48-unit NN has 24, 48, 48, and 24 units from the input layer to the output layer. Fig. 3 depicts the average MCD obtained from each method, showing that the wider architecture (such as "72") does not always provide better results than narrower architectures except when our method is used. RNN performed poorly, because it is based on random-initialization, while our method performed well due to our hierarchical learning algorithm.

For the remaining experiments in this paper, the best number of hidden units for each method were used; i.e., $J = 24$ for "SD-RBM" and "NN," $J = 48$ for "RNN," and $J = 72$ for "SD-RTRBM."

C. GMM-Based Methods

For GMM-based voice conversion, we tried and evaluated five mixtures (8, 16, 32, 64, 128 mixtures) to determine an appropriate number of mixtures. We further compared three types

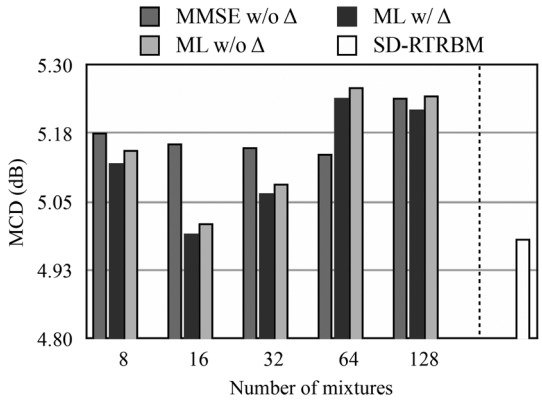

Fig. 4. Average MCD with varying numbers of mixtures for GMM methods. GMM on MMSE criterion without delta (MMSE w/o Δ), GMM on ML criterion without delta (ML w/o Δ), and GMM on ML criterion with delta (MMSE w/ Δ).

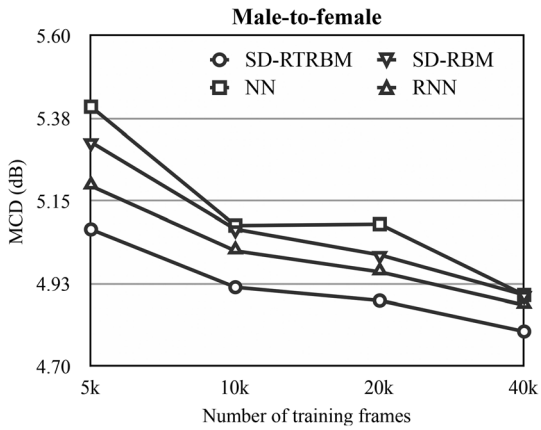

Fig. 5. Male-to-female voice conversion results. The values show average MCD for each method with varying amounts of training data.

of GMM-based VC methods: GMM on MMSE (minimum mean square error) without delta features, GMM on ML (maximum likelihood) criterion without delta features, and GMM on ML criterion with delta features. Fig. 4 shows the average MCDs over the development set when using the GMM with various mixtures. As shown in the figure, GMM on MMSE criterion without delta, GMM on ML without delta, GMM on ML with delta performed the best when 64, 16, and 16 mixtures were used, respectively. Specifically, GMM on ML criterion with delta performed best among the three GMM methods. When we compared the best GMM-based method ("ML w/ Δ with 16 mixtures) with our proposed method, the proposed method performed better (MCD of 4.98 by "SD-RTRBM" to MCD of 5.00 by "ML w/ Δ").

D. Changing the Number of Training Data

Figs. 5, 6, and 7 summarize the experimental results for the test set, comparing objective criteria for male-to-female, male-to-male, and female-to-female voice conversion, respectively, for $F = 5,000$, 10,000, 20,000, and 40,000 training frames. These figures also include the RNN results for reference. Table I shows the experimental results with respect to subjective criteria. Structure-tuned models are used for each network-based method in the subjective evaluation: 72 hidden units for SD-RTRBM, and 24 hidden units for SD-RBM and NN (see Section IV-B: *Network-based methods*). Since, in our preliminary objective evaluation, GMM with delta features on

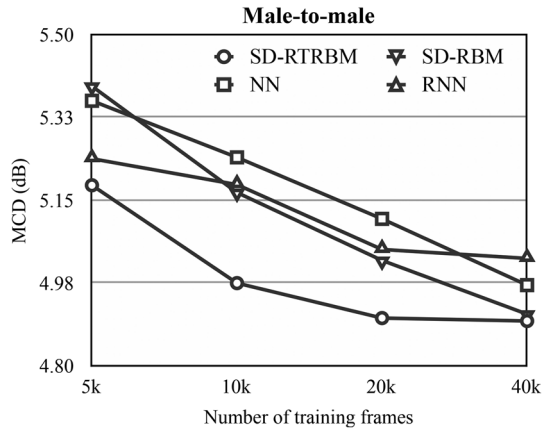

Fig. 6. Male-to-male voice conversion results. The values show average MCD for each method with varying amounts of training data.

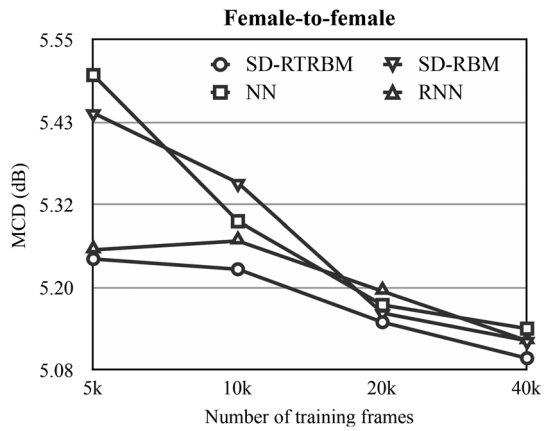

Fig. 7. Female-to-female voice conversion results. The values show average MCD for each method with varying amounts of training data.

TABLE I
AVERAGE MOS W.R.T. SIMILARITY FOR EACH METHOD. THE VALUES AFTER $\pm$ SYMBOL INDICATE THE CONFIDENCE INTERVALS

SD-RTRBM	SD-RBM	NN	GMM(ML,Δ)
3.444 $\pm$ 0.169	3.044 $\pm$ 0.163	2.911 $\pm$ 0.173	3.044 $\pm$ 0.194

ML criterion outperformed GMM without delta features on MMSE, we compared our method to GMM with delta on ML (IV-C: *GMM-based methods*).

As shown in these figures and the table, our approach (SD-RTRBM) outperformed the other methods in every case (regardless of the gender). The reason for the improvement is attributable to the fact that our time-dependent high-order conversion system using RTRBMs is able to capture and convert the abstractions of unique speaker characteristics better than the other methods. In particular, as shown in Figs. 5, 6, and 7, our approach achieved high performance in MCD criteria. This is because the RTRBMs modeled and captured sequence data more appropriately than the other methods and reduced estimation errors.

One interesting point is that most of the MCDs shown in Figs. 5, 6, and 7 decreased as the amount of training data increased, but the MCD of NN at $F = 20,000$ in the male-to-female experiment and the MCD of RNN at $F = 10,000$ in the female-to-female experiment increased with increases in training data. This is caused by a fall into local minima starting from the

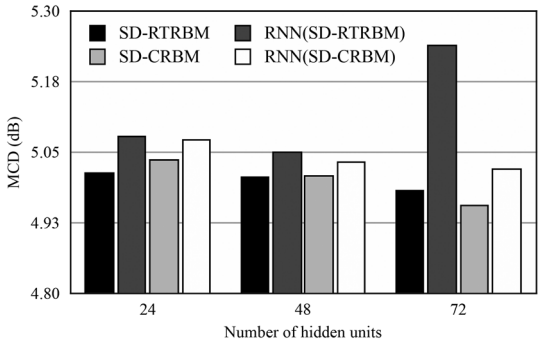

Fig. 8. Comparison between SD-RTRBM and SD-CRBM.

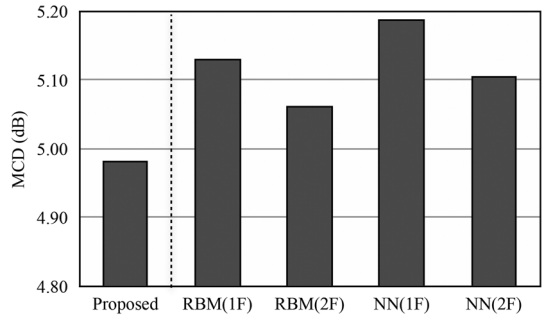

Fig. 9. Comparison of the proposed method and the conventional methods with segment features. "1F" and "2F" indicate that the model feeds one-frame features and two-frame features (segment features), respectively.

randomly-initialized weights. Our method provided more stable performance than the randomly-initialized methods.

E. SD-RTRBM vs. SD-CRBM

We further investigated how the proposed method works compared with our previous approach, called speaker-dependent conditional restricted Boltzmann machines (SD-CRBM) [42]. The SD-CRBM model tries to capture time-related information in time-series data, which is actually very similar to the proposed method. The difference is that the proposed method models temporal characteristics in hidden features, while SD-CRBM models in acoustic features directly.

The average MCD of the proposed method ("SD-RTRBM") and SD-CRBM over five sentences in the development set with varying numbers of hidden units are shown in Fig. 8. SD-CRBM feeds two 24-dimensional MFCC vectors of the source speaker and an estimated previous MFCC vector of the target speaker. For reference, we also compared each method with a randomly-initialized RNN that has the same architecture as it does, where two randomly-initialized RNNs that have the same architecture as SD-RTRBM and SD-CRBM are named "RNN(SD-RTRBM)" and "RNN(SD-CRBM)", respectively. As shown in Fig. 8, the proposed method and SD-CRBM produced similar results. Although SD-CRBM outperformed SD-RTRBM when 72 hidden units were used, our method has an advantage in that it feeds no more than one frame feature (while SD-CRBM requires a few frame-features; i.e., segment features). We can say that the proposed method performed as well as SD-CRBM, even though it feeds fewer input features.

F. Comparing with Segment Features

It might be interesting to compare our method, which actually feeds one-frame features, but propagates previously-estimated values in the hidden layers, with the conventional methods (speaker-dependent RBMs and NNs) that feed two adjacent acoustic features (segment features). Fig. 9 compares the average MCD obtained from the proposed method, speaker-dependent RBMs and random-initialized NN with segment features ("RBM(2F)" and "NN(2F)", respectively). The hyper-parameters of each model, such as the number of hidden units, are tuned. As shown in Fig. 9, we noticed that the proposed method outperformed the other methods with segment features, although our method feeds one-frame features. This indicates that the model that structurally considers time-related dependencies in the network produces better results than the model that simply feeds multiple features to capture time-related information.

V. Conclusions

In this paper, we presented a voice conversion method that combines speaker-dependent RTRBMs and a NN to extract time-dependent unique speaker information from sequence data. Using experiments, we confirmed that our approach is more effective, regardless of gender and especially in terms of MCD, than a well-known approach using a Gaussian mixture model (GMM), an approach based on a neural network (NN), and our own previous work on speaker-dependent restricted Boltzmann machines (SD-RBMs). The proposed method can be seen as a special way of generative initialization (training) of a recurrent neural network (RNN). Experimental results that compared our method and a random-initialized RNN indicated that the initial values of the parameter are fairly important for the improvement of accuracy and for stable training. In the future, we will further investigate these relationships and improve our method.

References

[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1998, pp. 285–288.
[2] C. Veaux and X. Robet, "Intonation conversion from neutral to expressive speech," in *Proc. Interspeech*, 2011, pp. 2765–2768.
[3] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using GMM-based voice conversion for electrolaryngeal speech," *Speech Commun.*, vol. 54, no. 1, pp. 134–146, 2012.
[4] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2001, pp. 301–304.
[5] A. Kunikoshi, Y. Qiao, N. Minematsu, and K. Hirose, "Speech generation from hand gestures based on space mapping," in *Proc. Interspeech*, 2009, pp. 308–311.
[6] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1988, pp. 655–658.
[7] H. Valbret, E. Moulines, and J.-P. Tubach, "Voice transformation using PSOLA technique," *Speech Commun.*, vol. 11, no. 2, pp. 175–187, 1992.
[8] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1998.
[9] C.-H. Lee and C.-H. Wu, "MAP-based adaptation for speech conversion using adaptation data selection and non-parallel training," in *Proc. Interspeech*, 2006, pp. 2254–2257.
[10] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on gaussian mixture model," in *Proc. Interspeech*, 2006, pp. 2446–2449.

[11] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Proc. Interspeech*, 2011, pp. 653–656.
[12] D. Saito, S. Watanabe, A. Nakamura, and N. Minematsu, "Probabilistic integration of joint density model and speaker model for voice conversion," in *Proc. Interspeech*, 2010, pp. 1728–1731.
[13] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2009, pp. 3893–3896.
[14] Y.-J. Wu, H. Kawai, J. Ni, and R.-H. Wang, "Minimum segmentation error based discriminative training for speech synthesis application," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2004, pp. 629–632.
[15] E. McDermott, T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large-vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, Jan. 2007.
[16] T. Tomoki and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. 90, no. 5, pp. 816–824, 2007.
[17] Z.-H. Ling and L.-R. Dai, "Minimum kullback–leibler divergence parameter generation for HMM-based speech synthesis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 5, pp. 1492–1502, Jul. 2012.
[18] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC system for blizzard challenge 2006. An improved HMM-based speech synthesis method," in *Proc. Blizzard Challenge Workshop*, 2006.
[19] Z. Jian and Z. Yang, "Voice conversion using canonical correlation analysis based on gaussian mixture model," in *Proc. 8th ACIS Int. Conf. IEEE Software Eng., Artif. Intell., Netw., Parallel/Distrib. Comput. (SNPD '07)*, 2007, pp. 210–215.
[20] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2012, pp. 313–317.
[21] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, "Exemplar-based voice conversion using non-negative spectrogram deconvolution," in *Proc. 8th ISCA Speech Synth. Workshop*, 2013.
[22] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *Proc. Interspeech*, 2013, pp. 369–372.
[23] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," *Parallel Distrib. Process.*, vol. 1, 1986.
[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[25] Z.-H. Ling, L. Deng, and D. 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, Oct. 2013.
[26] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.
[27] V. Nair and G. Hinton, "3-d object recognition with deep belief nets," *Adv. Neural Inf. Process. Syst.*, vol. 22, pp. 1339–1347, 2009.
[28] T. Deselaers, S. Hasan, O. Bender, and H. Ney, "A deep learning approach to machine transliteration," in *Proc. 4th Workshop Statist. Mach. Translat.. Assoc. Comput. Linguist.*, 2009, pp. 233–241.
[29] I. Sutskever, G. Hinton, and G. Taylor, "The recurrent temporal restricted Boltzmann machine," *NIPS*, vol. 19, pp. 1601–1608, 2008.
[30] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proc. Int. Conf. Mach. Learn.*, 2012.
[31] Z. Wu, E. S. Chng, and H. Li, "Conditional restricted Boltzmann machine for voice conversion," in *Proc. IEEE China Summit and Int. Conf. Signal Inf. Process. (ChinaSIP)*, 2013, pp. 104–108.
[32] L.-H. Chen, Z.-H. Ling, Y. Song, and L.-R. Dai, "Joint spectral distribution modeling using restricted Boltzmann machines for voice conversion," in *Proc. Interspeech*, 2013, pp. 3052–3056.
[33] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," *Adv. Neural Inf. Process. Syst.*, pp. 1345–1352, 2006.
[34] Y. Freund and D. Haussler, *Unsupervised Learning of Distributions of Binary Vectors Using Two Layer Networks*. Santa Cruz, CA, USA: Computer Research Laboratory, , 1994.
[35] K. Cho, A. Ilin, and T. Raiko, "Improved learning of gaussian-bernoulli restricted Boltzmann machines," *Artif. Neur. Netw. Mach. Learn.*, pp. 10–17, 2011.
[36] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2012.
[37] 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, Nov. 2007.
[38] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR japanese speech database as a tool of speech recognition and synthesis," *Speech Commun.*, vol. 9, no. 4, pp. 357–363, 1990.
[39] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 3933–3936.
[40] B. Milner and X. Shao, "Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model," in *Proc. Interspeech*, 2002, pp. 2421–2424.
[41] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Comput. Sci. Dept., Univ. of Toronto, Toronto, ON, USA, Tech. Rep., 2009.
[42] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion in time-invariant speaker-independent space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 7939–7943.

Toru Nakashika received his B.E. and M.E. degrees in computer science from Kobe University in 2009 and 2011, respectively. From September 2011 to August 2012, he studied at INSA de Lyon in France. In the same year, he continued his research as a doctoral student, and received his Dr.Eng. degree in computer science from Kobe University in 2014. He is currently an Assistant Professor at Kobe University. His research interest is speech and image recognition and statistical signal processing. He is a member of IEEE, ISCA and ASJ.

Tetsuya Takiguchi received his B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and his M.E. and Dr.Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively. From 1999 to 2004, he was a researcher at IBM Research, Tokyo Research Laboratory, Kanagawa, Japan. He is currently an Associate Professor at Kobe University. His research interests include statistic signal processing and pattern recognition. He received the Awaya Award from the Acoustical Society of Japan in 2002. He is a member of the IEEE, the IPSJ, and the ASJ.

Yasuo Ariki received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976, and 1979, respectively. He was an Assistant Professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an Associate Professor and from 1992 to 2003 a Professor at Ryukoku University. Since 2003, he has been a Professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database. He is a member of IEEE, IEICE, IPSJ, JSAI, ASJ, ITE, and IIEEJ.