INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Voice conversion based on trajectory model training of neural networks considering global variance

Naoki Hosaka, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Scientific and Engineering Simulation,
Nagoya Institute of Technology, Nagoya, Japan

Abstract

This paper proposes a new training method of deep neural networks (DNNs) for statistical voice conversion. DNNs are now being used as conversion models that represent mapping from source features to target features in statistical voice conversion. However, there are two major problems to be solved in conventional DNN-based voice conversion: 1) the inconsistency between the training and synthesis criteria, and 2) the over-smoothing of the generated parameter trajectories. In this paper, we introduce a parameter trajectory generation process considering the global variance (GV) into the training of DNNs for voice conversion. A consistent framework using the same criterion for both training and synthesis provides better conversion accuracy in the original static feature domain, and the over-smoothing can be avoided by optimizing the DNN parameters on the basis of the trajectory likelihood considering the GV. Experimental results show that the proposed method outperforms the DNN-based method in term of both speech quality and speaker similarity.

Index Terms: Voice conversion, statistical model, neural network, trajectory model, global variance

1. Introduction

Voice conversion is a technique for converting a certain speaker's voice into another speaker's voice. This conversion can modify nonlinguistic information such as voice characteristics while keeping linguistic information the same. One typical spectral conversion framework is based on a Gaussian mixture model (GMM) [1]. This method realizes a continuous mapping on the basis of soft clustering and converts spectral parameters frame-by-frame on the basis of the minimum mean square error.

Deep neural networks (DNNs), which are feed-forward artificial neural networks with many hidden layers, have recently achieved significant improvement in automatic speech recognition [2]. DNNs have also been applied to voice conversion, where they represent complex mapping functions from acoustic features for source speech to ones for target speech. State-of-the-art voice conversion based on DNN [3] achieved accurate conversion and high speech quality by directly transforming high-dimensional spectral features. DNN-based voice conversion shows the potential to produce more natural-sounding voice conversion.

This paper focuses on two problems in DNN-based voice conversion: 1) inconsistency between the training and synthesis criteria, and 2) over-smoothing of the generated parameter trajectories. In the training process of DNNs, a frame-by-frame independence is generally assumed and frame-level objective functions, such as the mean squared error between converted and target features, are widely used to train DNNs. On the other hand, in the synthesis process, a parameter generation algorithm using dynamic features is generally used, and static feature sequences are generated considering the relationship between neighboring static features, i.e., objective functions with respect to static feature sequences are used to generate smooth parameter trajectories. Consequently, the training and synthesis criteria and inconsistent, and DNNs cannot be optimized for parameter generation. In addition, the static feature vectors generated by the conventional generation process are usually over-smoothed, and this is one of the main factors causing the muffled effect in statistical voice conversion. For improving the converted speech quality, Toda and Tokuda [4] introduced a new criterion on a higher order moment called the global variance (GV), which is the variance of the static feature vectors calculated over a time sequence (e.g. over an utterance), into the parameter generation process. The parameter generation considering GV is widely used in statistical voice conversion and speech synthesis, and it has been reported that quality of speech can be significantly improved by generating the parameter trajectory while keeping its GV close to the natural one [4, 5, 6]. To address these problems, we introduce a trajectory training method considering the GV proposed for HMM-based speech synthesis [7] and DNN-based speech synthesis [8]. Recent works in DNN-based speech synthesis using a trajectory training considering the GV have reported that the method outperformed the conventional DNN-based method in terms of the naturalness of synthesized speech [8].

This paper introduces the trajectory training method considering the GV into DNN-based voice conversion. DNNs can be optimized for converting source feature trajectories into target feature trajectories in the sense of maximum likelihood subject to a constraint on the GV of the converted parameter trajectory. Consequently, a unified framework that consistently uses the same criterion in both training and synthesis is obtained, and the over-smoothing problem is alleviated. In this paper, we evaluate the effectiveness of the proposed trajectory training method on objective and subjective measures. Experimental results show that the proposed method significantly improved on the conventional DNN-based method in terms of speaker similarity.

The rest of this paper is organized as follows. Sections 2 and 3 describe voice conversion based on DNNs and the proposed training method, respectively. The experimental conditions and results are given in Section 4. Concluding remarks and future work are presented in Section 5.

2. Voice conversion using neural networks

In voice conversion using neural networks (NN) [9], a NN is trained to represent a mapping function from source features to target features consisting of spectral features with their dynamic features. In the generation process, target features are obtained

http://dx.doi.org/10.21437/Interspeech.2016-1035

from given source features by the trained DNN using forward propagation. Although static target features can be generated directly by the DNN, the speech parameter trajectories generated by the parameter generation algorithm considering the explicit relationship between static and dynamic features have been reported to perform better [6] in the field of voice conversion. Therefore, in this work, the parameter generation is applied for generating smooth speech parameter trajectories.

A target feature vector $\boldsymbol{Y}_t$ consists of a D-dimensional static feature vector $\boldsymbol{y}_t = [y_t(1), \ldots, y_t(D)]^\top$ and their dynamic feature vector.

$$\boldsymbol{Y}_t = [\boldsymbol{y}_t^\top, \Delta^{(1)}\boldsymbol{y}_t^\top]^\top \tag{1}$$

The target feature vector sequence $\boldsymbol{Y}$ and the static feature vector sequence $\boldsymbol{y}$, which represent an utterance, can be written in vector forms as follows

$$\boldsymbol{Y} = [\boldsymbol{Y}_1^\top, \ldots, \boldsymbol{Y}_t^\top, \ldots, \boldsymbol{Y}_T^\top]^\top \tag{2}$$

$$\boldsymbol{y} = [\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_t^\top, \ldots, \boldsymbol{y}_T^\top]^\top \tag{3}$$

where T is the number of frames included in an utterance. The relationship between $\boldsymbol{Y}$ and $\boldsymbol{y}$ can be represented by $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{y}$, where $\boldsymbol{W}$ is a window matrix extending the static feature vector sequence $\boldsymbol{y}$ to the target feature vector sequence $\boldsymbol{Y}$. The optimal static feature vector sequence is obtained by

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{y}} \mathcal{N}(\boldsymbol{W}\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \bar{\boldsymbol{y}} \tag{4}$$

where $\boldsymbol{\lambda}$ is a parameter set and $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_t^\top, \ldots, \boldsymbol{\mu}_T^\top\right]^\top \tag{5}$$

$$\boldsymbol{\Sigma} = \text{diag}\left[\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_t, \ldots, \boldsymbol{\Sigma}_T\right] \tag{6}$$

The optimal static feature sequence $\hat{\boldsymbol{y}}$ is given by

$$\hat{\boldsymbol{y}} = \left(\boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} = \boldsymbol{P}\boldsymbol{r} \tag{7}$$

where

$$\boldsymbol{P} = \left(\boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{W}\right)^{-1}, \qquad \boldsymbol{r} = \boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \tag{8}$$

As a result, smooth static feature trajectories can be obtained by using dynamic features as constraints.

In DNN-based voice conversion, the mean vector at frame t, $\boldsymbol{\mu}_t$, is obtained from a trained neural network and a source feature vector at time t, $\boldsymbol{x}_t$, as follows:

$$\boldsymbol{\mu}_t = g(\boldsymbol{x}_t|\boldsymbol{\lambda}_{NN}) \tag{9}$$

where $g(\cdot|\boldsymbol{\lambda}_{NN})$ is a non-linear mapping function represented by a neural network $\boldsymbol{\lambda}_{NN}$. A covariance matrix is usually independent of linguistic features, i.e., a globally tied covariance matrix $\boldsymbol{\Sigma}_g$ is used, in DNN-based voice conversion.

Assuming that outputs of a neural network are used as mean parameters in a statistical model, an objective function can be defined as

$$\mathcal{L} = P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{Y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{Y}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_g) \tag{10}$$

The parameter set $\boldsymbol{\lambda}$, which consists of the parameter of the neural network and the covariance matrix $\boldsymbol{\Sigma}_g$, is optimized in the sense of maximum likelihood as follows:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{Y}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_g) \tag{11}$$

If an identity matrix is used as the covariance matrix $\boldsymbol{\Sigma}_g$, maximization of the objective function $\mathcal{L}$ is equivalent to minimization of the conventional frame-level mean square errors. The neural network can be trained by standard back-propagation using the gradient of the mean vector.

3. Trajectory training method considering global variance for DNNs

3.1. Trajectory training

In the conventional DNN-based voice conversion framework, although the frame-level objective function is used for DNN training, the sequence-level objective function is used for parameter generation. To address this inconsistency between training and synthesis, a trajectory training method is introduced into the training process of DNNs.

The conventional likelihood function in Eq. (10) can be reformulated as a trajectory likelihood function by imposing explicit relationship between static and dynamic features, which is given by $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{y}$ [10]. The trajectory likelihood function of $\boldsymbol{y}$ is then written as

$$\mathcal{L}_{Trj} = \frac{1}{Z} P(\boldsymbol{Y}|\boldsymbol{\lambda}) = P(\boldsymbol{y}|\boldsymbol{\lambda}) = \mathcal{N}\left(\boldsymbol{y} \mid \bar{\boldsymbol{y}}, \boldsymbol{P}\right) \tag{12}$$

where Z is a normalization term. Inter-frame correlation is modeled by the covariance matrix $\boldsymbol{P}$ that is generally full. Note that the mean vector $\bar{\boldsymbol{y}}$ is equivalent to the generated static feature sequence shown by Eq. (7).

The parameter set $\boldsymbol{\lambda}$ is estimated by maximizing the trajectory likelihood $\mathcal{L}_{Trj}$. The gradients of mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ can be calculated as follows

$$\frac{\partial \mathcal{L}_{Trj}}{\partial \boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1} \boldsymbol{W} \left(\boldsymbol{y} - \bar{\boldsymbol{y}}\right) \tag{13}$$

$$\begin{aligned} \frac{\partial \mathcal{L}_{Trj}}{\partial \boldsymbol{\Sigma}^{-1}} =& \frac{1}{2}\text{diag}\Big[\boldsymbol{W}\left(\boldsymbol{P} - \boldsymbol{y}\boldsymbol{y}^\top + \bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\top\right)\boldsymbol{W}^\top \\ & - 2\boldsymbol{\mu}\left(\bar{\boldsymbol{y}} - \boldsymbol{y}\right)^\top \boldsymbol{W}^\top\Big] \end{aligned} \tag{14}$$

The parameters of neural network are updated by the back-propagation algorithm using the gradient in Eq. (13). The computation of gradients for the parameters of the neural network in lower layers is the same as the calculation of gradients for standard neural networks. The covariance matrix $\boldsymbol{\Sigma}$ is iteratively updated by using the gradient in Eq. (14).

3.2. Trajectory training considering GV

A chart of trajectory training considering GV is shown in Fig. 1. To address the over-smoothing problem of generated parameter trajectories, the concept of parameter generation considering the GV is introduced into the training of DNNs. The proposed objective function $\mathcal{L}_{GVTrj}$ is given by

$$\begin{aligned} \mathcal{L}_{GVTrj} &= P(\boldsymbol{y}|\boldsymbol{\lambda}) P(\boldsymbol{v}(\boldsymbol{y})|\boldsymbol{\lambda}, \boldsymbol{\lambda}_v)^{\omega T} \\ &= \mathcal{N}(\boldsymbol{y} \mid \bar{\boldsymbol{y}}, \boldsymbol{P}) \mathcal{N}\left(\boldsymbol{v}(\boldsymbol{y}) \mid \boldsymbol{v}(\bar{\boldsymbol{y}}), \boldsymbol{\Sigma}_v\right)^{\omega T} \end{aligned} \tag{15}$$

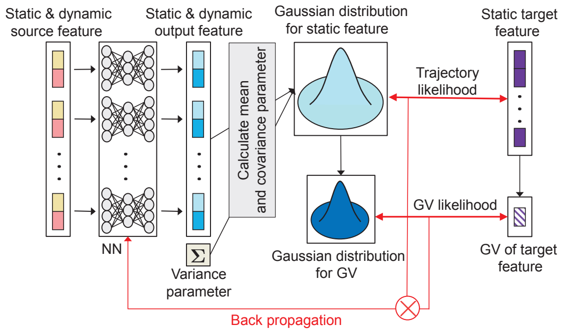
Static & dynamic source feature
Static & dynamic output feature
Gaussian distribution for static feature
Static target feature
Calculate mean and covariance parameter
Trajectory likelihood
NN
Σ
Variance parameter
Gaussian distribution for GV
GV likelihood
GV of target feature
Back propagation

Figure 1: Chart of trajectory training considering GV

where $\boldsymbol{v}(\boldsymbol{y}) = [v(1), \dots, v(D)]^\top$ is a GV vector of the static feature vector sequence $\boldsymbol{y}$. The globally shared covariance matrix $\boldsymbol{\Sigma}_v$ is independent from the input feature. The NN, the Gaussian distribution $\boldsymbol{\lambda}_v$, and the NN $\boldsymbol{\lambda}$ are concurrently trained using the training data. The GV vector is calculated utterance by utterance as follows:

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}(y_t(d) - \langle y(d)\rangle)^2 \tag{16}$$

$$\langle y(d)\rangle = \frac{1}{T}\sum_{t=1}^{T} y_t(d) \tag{17}$$

where d is an index of the feature dimension. The mean vector of the probability density for the GV, $\boldsymbol{v}(\bar{\boldsymbol{y}})$, is defined as the GV of the mean vector of the trajectory likelihood function in Eq. (12), which is equivalent to the GV of the generated parameters shown by Eq. (7). The GV likelihood $P(\boldsymbol{v}(\boldsymbol{y})|\boldsymbol{\lambda}, \boldsymbol{\lambda}_v)$ works as a penalty term to make the GV of the generated parameters close to that of the natural ones. The balance between the two likelihoods $P(\boldsymbol{y}|\boldsymbol{\lambda})$ and $P(\boldsymbol{v}(\boldsymbol{y})|\boldsymbol{\lambda}, \boldsymbol{\lambda}_v)$ is controlled by the GV weight w.

The parameter set, which consists of the parameter of the neural network and the covariance matrix $\boldsymbol{\Sigma}$, is estimated by maximizing the proposed objective function $\mathcal{L}_{GVTrj}$. The gradients of the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ can be calculated as follows:

$$\frac{\partial \mathcal{L}_{GVTrj}}{\partial \boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\boldsymbol{y} - \bar{\boldsymbol{y}} + w\boldsymbol{P}\bar{\boldsymbol{x}}) \tag{18}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{GVTrj}}{\partial \boldsymbol{\Sigma}^{-1}} =& \frac{1}{2}\mathrm{diag}\big[\boldsymbol{W}(\boldsymbol{P} + \bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\top - \boldsymbol{y}\boldsymbol{y}^\top)\boldsymbol{W}^\top \\ &- 2\boldsymbol{\mu}(\bar{\boldsymbol{y}} - \boldsymbol{y})^\top\boldsymbol{W}^\top + 2w\boldsymbol{W}\boldsymbol{P}\bar{\boldsymbol{x}}(\boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{y})^\top\big]\end{aligned} \tag{19}$$

$$\bar{\boldsymbol{x}} = -2\boldsymbol{P}_v(\bar{\boldsymbol{y}} - \langle\bar{\boldsymbol{y}}\rangle) \tag{20}$$

$$\boldsymbol{P}_v = \mathrm{diag}\left[\boldsymbol{I}_{T\times T} \otimes \left(\boldsymbol{\Sigma}_v^{-1}(\boldsymbol{v}(\bar{\boldsymbol{y}}) - \boldsymbol{v}(\boldsymbol{y}))\right)\right] \tag{21}$$

where $\otimes$ is a Kronecker product, and $\langle\bar{\boldsymbol{y}}\rangle$ is the mean of $\bar{\boldsymbol{y}}$. The neural network can be updated and trained by the backpropagation algorithm using the gradient in Eq. (18). The computation of gradients for the parameters in lower layers is the same as the calculation of gradients for standard neural networks. The parameters are optimized so that the GVs of generated trajectories move close to the natural ones.

The optimal static feature vector sequence is determined by maximizing the objective function $\mathcal{L}_{GVTrj}$ as follows:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{\lambda})P(\boldsymbol{v}(\boldsymbol{y})|\boldsymbol{\lambda}, \boldsymbol{\lambda}_v) \tag{22}$$

Since this estimate is equivalent to the ML estimate by the basic parameter generation algorithm shown by Eq. (4), the basic parameter generation algorithm can be employed for the proposed framework. Note that the basic algorithm is computationally much more efficient than the parameter generation algorithm considering the GV [4] that requires an iterative process. In addition, all frames of target vector sequences are used for dynamic time warping (DTW) between the source and target vector sequences in the trajectory method.

4. Experiments

4.1. Experimental conditions

A Japanese speech database, which was constructed by our research group, was used for this experiment. The database contains sets of 503 phonetically balanced sentences uttered by more than 100 college students. The contents of the data are the same as the B-set of the ATR phonetically balanced Japanese speech database [11]. We selected a set of source and target male speakers. The source speaker was m002, and the target one was m001. Two training sets consisting of 10 and 450 sentences were used for training, and the remaining 53 sentences were used for evaluation. Speech signals were sampled at 48 kHz. Feature vectors were extracted with a 5-ms shift and the feature vector consisted of the 0-th through 49-th mel-cepstral coefficients.Mel-cepstral coefficients were extracted from the smoothed spectrum analyzed by STRAIGHT [12]. In these experiments, the following four systems were compared.

- **GMM**: Conventional GMM-based voice conversion system
- **DNN**: Voice conversion system based on DNN trained by maximizing the objective function in Eq. (10)
- **TrjDNN**: Voice conversion system based on DNN trained by maximizing the objective function in Eq. (12)
- **GVTrjDNN**: Voice conversion system based on DNN trained by maximizing the objective function in Eq. (15)

In **GMM**, the number of mixture components was set to 32, and the covariance matrices were full. Trajectory training and GV training were not used for GMM. The source and target features were 100-dimensional acoustic feature vectors, consisting of 50 mel-cepstral coefficients and their dynamic features (delta). The GMMs were trained with the EM algorithm using the joint vectors, which were aligned by DTW, in a training set. In DNN-based systems, the source and target features were normalized to have zero-mean unit-variance, respectively. DTW was applied to obtain optimal frame alignment for training the DNNs. For **TrjDNN** and **GVTrjDNN**, DTW was performed with the constraints that target features are not skipped and not duplicated, i.e., feature sequence aligned to target duration. The architectures of the DNNs used in all DNN-based systems were 3-hidden-layer with 1024 units per layer for the large training set and 4-hidden-layer with 256 units per layer for the small training set. The sigmoid activation function was used in the hidden layers, and the linear activation function was used in the output layer. The weights of the DNN used in **DNN** were initialized randomly and then optimized by maximizing the objective function $\mathcal{L}$ in Eq. (10). The trained DNN was used as an initial model for TrjDNN. The initial model for GVTrjDNN was the DNN trained in TrjDNN. For training the DNNs, a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm was used. For **TrjDNN** and **GVTrjDNN**, utterance-level batches were used in the SGD-based training, i.e., an utterance was used as a mini-batch in SGD-based training. The

Table 1: Global variance distances and Mel-cepstral distortions (dB) on test data (450 training data sets).

	GMM	DNN	TrjDNN	GVTrjDNN
GVD	0.379	0.430	0.378	0.349
MCD	4.579	4.435	4.426	4.429

GV weights were set to 0.025 for the large training set and 0.05 for the small training set, respectively[1]. The basic parameter generation algorithm was applied to generate parameter trajectories for all systems. To measure only the performance of the spectral conversion, F_0 was converted using the conventional method, which is simply a linear transformation in the log-scale to equalize the mean and variance of the converted and target speech samples.

4.2. Experimental results

To objectively evaluate the performance of the systems, the GV distance (GVD) for mel-cepstrum coefficients and the mel-cepstral distortion (MCD) were used as objective measures. The GVD was calculated by

$$\mathrm{GVD} = \sqrt{\frac{\sum_{n=1}^{N} \sum_{d=1}^{D} \left(v_n\left(d\right) - \bar{v}_n\left(d\right)\right)^2}{N}} \tag{23}$$

where N is the number of test data and D is the number of dimensions of mel-cepstral coefficients. Tables 1 and 2 list the objective evaluation results. In both training conditions, **GVTrjDNN** achieved significantly lower GVD than **DNN**. Additionally, **GVTrjDNN** even achieved lower GVD than from **TrjDNN**. These results show that the over-smoothing problem was alleviated by employing the trajectory training method considering the GV. Additionally, **TrjDNN** and **GVTrjDNN** outperformed **DNN** in the MCD evaluation in both training conditions. This result indicates that the trajectory training method strongly affects the conversion accuracy in voice conversion.

The Degradation Mean Opinion Score (DMOS) test was conducted for evaluating the similarity between the target and converted speech samples in terms of speaker characteristics. The opinion score was set on a five-point scale (5: imperceptible, 4: perceptible, but not annoying, 3:slightly annoying, 2: annoying, 1: very annoying). Fifteen sentences were selected randomly from test data for each subject. There were 10 subjects , who were all Japanese university students in our research group. Figure 2 and 3 show the results of the DMOS test. **GVTrjDNN** outperformed **GMM**, **DNN**, and **TrjDNN**, as shown in Fig. 2. **GVTrjDNN** scored higher than **TrjDNN**, though the difference between them was not statistically significant when 450 training data-sets were used. However, **GVTrjDNN** achieved significantly higher DMOS than **DNN**. These results indicate that the similarity of converted speech is drastically improved by introducing the parameter generation process into the training of DNNs. **DNN** had better MCD than **GMM** but worse, especially in the 10-dataset training condition. However, **GVTrjDNN** obtained similar GVD results in both training conditions while obtaining better MCD results. These results clearly show that the trajectory training considering GV is effective even if the training data is small.

[1] The GV weight was decided from preliminary experiments.

Table 2: Global variance distances and Mel-cepstral distortions (dB) on test data (10 training data sets).

	GMM	DNN	TrjDNN	GVTrjDNN
GVD	0.450	1.068	0.639	0.487
MCD	5.447	5.219	5.150	5.155

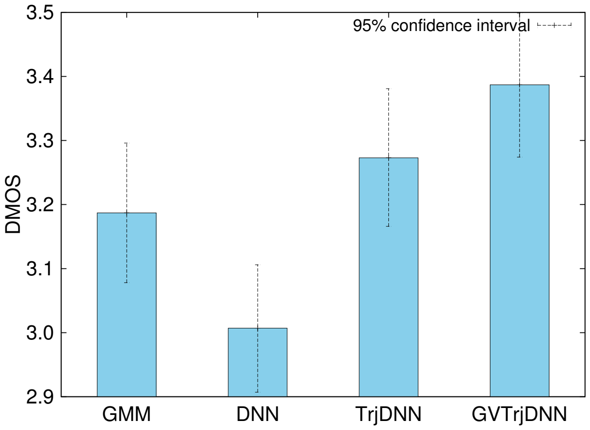

Figure 2: Degradation mean opinion score of the four conversion systems (450 training data sets).

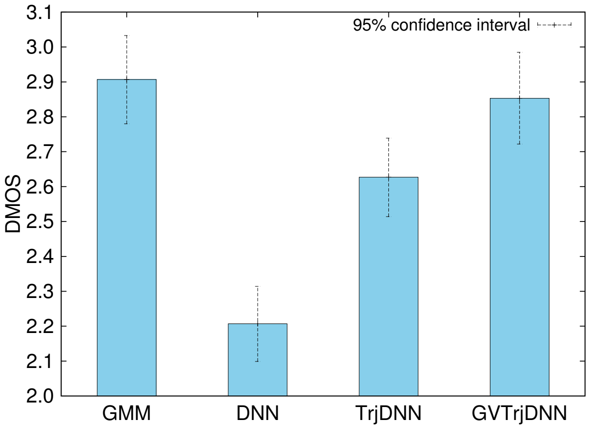

Figure 3: Degradation mean opinion score of the four conversion systems (10 training data sets).

5. Conclusions

In this paper, a trajectory training method considering the GV is proposed for DNN-based voice conversion. The proposed method solves the inconsistency between training and synthesis criteria and the over-smoothing problem. Experimental results show the proposed method can alleviate the over-smoothing problem and make converted speech more natural than that of a conventional DNN-based system. Future work will include some extensive experiments to compare the proposed method with the parameter generation method considering the GV and the other trajectory training methods [3, 13].

6. Acknowledgements

This work was supported by the Core Research for Evolutionary Science and Technology (CREST) program from the Japan Science and Technology Agency (JST).

7. References

[1] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous Probabilistic Transform for Voice Conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, pp. 131–142, 1998.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no.6, pp. 82–97, 2012.

[3] F. L. Xie, Y. Qian, Y. Fan, F. K. Soong, and H. Li, "Sequence error SE minimization training of neural network for voice conversion," *Proceedings of Interspeech 2014*, pp. 2283–2287, 2014.

[4] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information & Systems*, vol. E90-D, no. 5, pp. 816–824, 2007.

[5] M. Shannon and W. Byrne, "Fast, low-artifact speech synthesis considering global variance," *Proceedings of ICASSP 2013*, pp. 7869–7873, 2013.

[6] T. Toda, A. W. Black, and K. Tokuda, "Voice Conversion Based on Maximum Likelihood Estimation of Spectral Parameter Trajectory," *IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 8*, pp. 2222–2235, 2007.

[7] T. Toda and S. Young,"Trajectory training considering global variance for HMM-based speech synthesis," *Proceedings of ICASSP 2009*, pp. 4025–4028, 2009.

[8] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Trajectory training considering global variance for speech synthesis based on neural networks," *Proceedings of ICASSP 2016*, pp. 5600–5604, 2016.

[9] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," *Proceedings of ICASSP*, pp. 3893–3896, 2009.

[10] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequencesx," *Computer Speech and Language 21*, pp. 153–173, 2007.

[11] 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 Communication*, vol. 9, pp. 357–363, 1990.

[12] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[13] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," *Proceedings of Interspeech 2015*, pp. 309–313, 2015.