Singing Voice Conversion Using Posted Waveform Data on Music Social Media

Koki Senda*, Yukiya Hono*, Kei Sawada*, Kei Hashimoto*, Keiichiro Oura*, Yoshihiko Nankaku* and Keiichi Tokuda*
* Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan
E-mail: {kksnd924, hono, swdkei, bonanza, uratec, nankaku, tokuda}@sp.nitech.ac.jp Tel: +81-52-735-5479

***Abstract*—This paper proposes a method of selecting training data for many-to-one singing voice conversion (VC) from waveform data on the social media music app "nana." On this social media app, users can share sounds such as speaking, singing, and instrumental music recorded by their smartphones. The number of hours of accumulated waveform data has exceeded one million, and it is regarded as "big data." It is widely known that big data can create huge values by advanced deep learning technology. A lot of post data of multiple users having sung the same song is contained in nana's database. This data is considered suitable training data for VC. This is because VC frameworks based on statistical approaches often require parallel data sets that consist of pairs of waveform data of source and target singers who sing the same phrases. The proposed method can compose parallel data sets that can be used for many-to-one statistical VCs from nana's database by extracting frames that have small differences in the timing of utterances, based on the results of dynamic programming (DP) matching. Experimental results indicate that a system that uses training data composed by our method can convert acoustic features more accurately than a system that does not use the method.**

I. Introduction

Social media has made it possible for people all over the world to transmit their information. There are many kinds of social media websites and apps, such as YouTube, Facebook, and Instagram, and a large amount of data transmitted by users has been accumulating. This "big data" has increasing potential in creating values for every field[1]. Recently, machine learning methods of dealing with big data have been widely researched in many institutes and laboratories. The Multi-Genre Broadcast (MGB) Challenge—an official challenge of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)—is one of the international workshops evaluating big data technologies related to the speech field[3]. The challenge at ASRU 2015 was an evaluation of transcription[6], [7], speaker diarization[4], [5], dialect detection, and lightly supervised alignment using approximately 1,600 hours of British Broadcasting Corporation (BBC) television program data that had been recorded. Commercialized products using deep learning technology already exist, such as the speech recognition system employed in Google Home, which was trained using tens of thousands of hours of speech data[2].

The social media app "nana"[8] stores big data of music waveform data. This social media app is designed to allow users to share singing and instrumental sounds easily using their smartphone. Up to more than one million hours of waveform data have been uploaded, and the amount continues to increase. In nana, users can collaborate with other users' uploaded posts by overdubbing other user's sounds with their own sounds. In particular, accompaniment posts of popular songs are collaborated on by many users. The relationship between collaborating and collaborated posts is represented by a tree structure. Because each tree generally consists of one song, the same song is sung in almost all singing posts of each tree. A database that contains a large number of posts in which the same songs are sung by multiple users has large potential.

This data can be used for a parallel data set, the training data of voice conversion (VC). VC is a method of converting a speaker's voice into another kind of voice, especially another speaker's voice, while maintaining linguistic information. Statistical approaches have been widely researched[9], [10]. The conventional statistical VC is often based on a Gaussian mixture model (GMM) [11]. More recently, a VC framework based on deep neural networks (DNNs) has been proposed[12], [13]. These statistical VC typically train statistical models by using a parallel data set that consists of pairs of speech data from source and target speakers uttering the same sentences. Not every data can be used when the nana's waveform data is used for VC training data. For example, the database contains some partly sung data. We propose a method based on dynamic programming (DP) matching for composing training data from the database. The target data and the other data in the same collaboration tree are compared by DP matching, then a parallel data set is extracted.

The rest of this paper is organized as follows. Section 2 and 3 describe the social media music app nana and voice conversion using nana's waveform data, respectively. Section 4 describes the experimental conditions and experimental results. Section 5 presents concluding remarks and future work.

II. Social Media Music App "nana"

The social media music app nana[8] was developed by nana music, Inc. as a social music platform. The users can record and upload sounds such as speaking, singing, and instrumental music to nana with their smartphones. Through the app, users worldwide can communicate with each other through music. As of April 2018, there are six million users in 113 countries.

Fig. 1. The recording process.

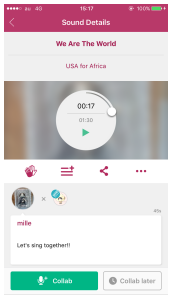

Fig. 2. Playing the sound of a post.

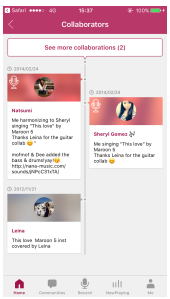

Fig. 3. Showing all the collaborators.

More than 61 million posts have accumulated in the database. Furthermore, the number of posts is still increasing.

The users upload sounds to nana according to the following procedure:

1) Record sounds.
2) Add information about recorded sounds, such as title, artist's name, and explanation.
3) Choose sound effects, such as echo, to arrange sounds.

Fig. 1 shows the screen of the recording process. Users can listen to uploaded posts, as shown in Fig. 2, and the users get feedback such as "Comment" and "Applause" (function equivalent to "Like") from other users.

In addition to these general functions, users can collaborate on posts. This characteristic function is called "Collab." The users can post their sounds, which are overdubbed on another user′s post. The function has the following two main effects. First, multiple users can create one sound together. For example, multiple users' singing is overdubbed to create choruses, and instruments are overdubbed to create band performances. Fig. 3 shows the relationship between collaborators on a post. On the screen shown in the figure, you can see all the posts in the "Collab" series of each post. Second, users can easily post accompanied singing voices because they can use an accompaniment post of another user. They do not have to prepare an accompaniment sound source by themselves; all they need to do is sing. Because many users have posted their singing in this way, the database holds many songs sung by multiple users. Popular songs are typically sung by tens of thousands of users.

The posts using "Collab" have two types of waveform data. The first type is mixed sound source data that all the posts in the "Collab" series are overdubbed by. The users are able to listen to only this type of sound source. The other type is single source data that consists of just a sound recorded when posting. In most cases, each data represents only one singing voice or one instrumental sound, although some of this type of data has multiple sounds; for example, singing with an instrument, such as guitar or piano, at the same time.

All the posts uploaded using "Collab" are related to the collaborated post. This relationship is represented by a tree structure representing each post as a node. When a post A exists and a post B collaborates with A, the post A becomes the parent node and the post B becomes the child node. Fig. 4 shows an example of the tree structure. The post that collaborated first becomes the root node and is accompaniment in most cases. In Fig. 4, the guitar post is the root node. Generally, every song tree is composed of singing voices and instrumental sounds related to one song. In almost all singing voice posts of each tree, the same song is sung because they have been sung with the same accompaniment. Fig. 5 shows an example of singing voice posts regarded as the same song in a tree.

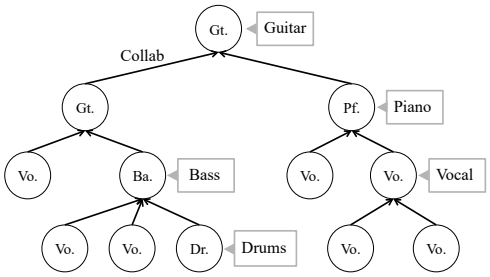

Fig. 4. A tree structure of an example collaboration relationship.

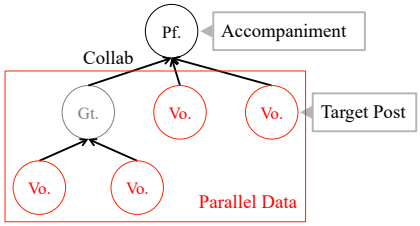

Fig. 5. The posts that are regarded as the same song because they have the same root node post.

We focus on this tree structure to extract such singing voice posts sung by many users for many-to-one singing voice conversion.

III. Voice Conversion Using Singing Post Data

Voice conversion (VC) is a method of converting an input speaker's voice into various types of voices while keeping linguistic information unchanged. This is mainly used for speaker conversion. A typical VC framework uses a statistical approach[9], [10]. In statistical VC, parallel data sets, which consist of pairs of speech data from source and target speakers uttering the same sentences, are used for training models. One conventional statistical VC is based on a Gaussian mixture model (GMM) [11]. GMM-based VC represents the relationship between acoustic features of a source and of a target speaker using linear combined multiple Gaussian distributions. A new approach based on deep neural networks (DNNs) has been proposed[12], [13]. This can convert acoustic features at a higher degree of precision than the GMM-based one. VC approaches are also distinguished by the number of source

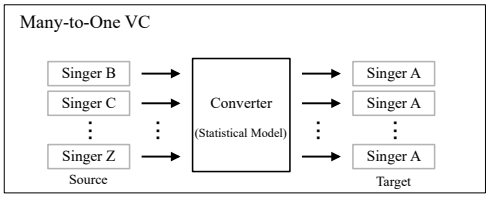

Fig. 6. Overview of many-to-one singing VC

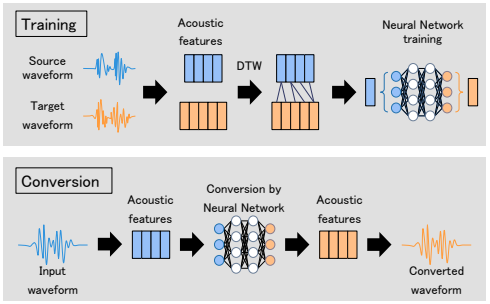

Fig. 7. Overview of our VC system

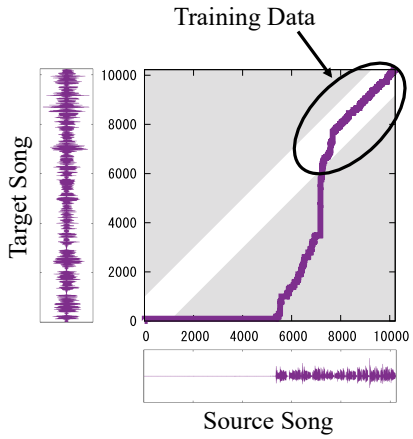

Fig. 8. [The training data extraction method that we propose.

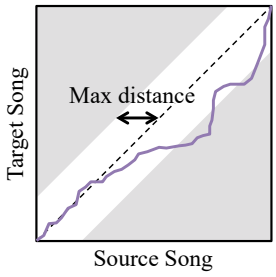

Fig. 9. Max distance of matching pass from diagonal.

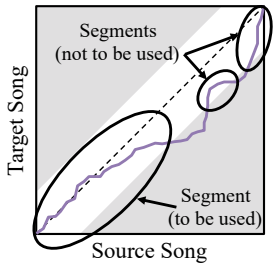

Fig. 10. Selection of training data considering segment length.

and target speakers. In addition, other approaches exist, such as singing VC. Examples also include conversion of sexuality, age, etc. We employ DNN-based many-to-one singing VC in our system, which converts an arbitrary singer's voice into a particular singer's voice.

In many-to-one singing VC, the input singing of an arbitrary singer (source singer) is converted into the singing of a particular singer (target singer), as shown in Fig. 6. Therefore, the parallel data set has to consist of multiple source singers' voices and one target singer's voice. Fig. 7 shows an overview of our VC system. In the training step, first, the acoustic features are extracted from source and target waveform data. Then, time alignment between these feature sequences are obtained by dynamic time wrapping (DTW)[14]. Finally, the neural network conversion model is trained using time-aligned acoustic feature sequences. In the conversion step, acoustic features extracted from input waveform data are converted by the trained model frame-by-frame. Then, the output singing voice is synthesized using a vocoder from converted features.

We extracted the singing data set of many users singing the same song from the nana′s database and applied it to this VC system because it is suitable for a parallel data set. Although it is generally difficult to get intended data from big data, such singing voice data is easily extracted from the database using tree structure representing collaboration relationships (Fig. 5). However, all of the extracted waveform data is not necessarily the same phrases, because users can record and post arbitrary content. For instance, there are some posts in which a singer is harmonizing with another singer and others sing only the hook of a song. Hence, a method is needed to remove unsuitable data and create appropriate parallel data sets.

An approach employing dynamic programming (DP) matching to an extract parallel data set has been proposed on the assumption that any two data of users singing the same song has a higher similarity than two other data of users singing different songs[15]. DP matching is a classical elastic matching method, widely applied to pattern recognition tasks such as speech recognition[14] and character recognition[16]. It can dynamically match each vector of two vector series that have different lengths, and the result is called a ″matching path.″ Then, the accumulation of Euclidean distances between matched vectors is calculated simultaneously at the end of matching. It indicates similarity between vector series. Therefore, it is possible to compare the similarities between two randomly selected post data in the database that have different lengths. In the conventional method, first, a target post is decided. Then, all the singing voice posts in the same tree are compared with the target post by DP matching, and the posts that have a small value of the accumulation of distances are extracted as source data of a parallel data set. However, in this method, most singers of the selected source data would be similar to the target singer because the accumulation of distances depends on the similarity between singers′ voices as well as what song was sung. In many-to-one singing VC, various types of voice data should be used for source training

data to convert arbitrary singers' voices.

Our method uses matching paths instead of the accumulation of distances. In our method, the differences in utterance timing between matched frames is calculated from matching paths, and the pairs of frames that have a smaller value of calculation results than threshold are extracted to be used for training based on the hypothesis that the posts that were sung with the same accompaniment have small differences in the utterance timing of each phrase. When satisfying the conditions of this method, the matching path is close to diagonal, as shown in Fig. 8. We call every part of the matching path extracted with this method a "segment." We expect to use our method to remove unsuitable data and create parallel data sets that consists of various types of voices. In this method, two parameters have to be set. The first is the maximum value of distance between matching paths and diagonal. We call this parameter the 〝max distance〞 (Fig. 9). Increasing this value increases the amount of data while degrading the quality of the data. The second one is the minimum value of segment length. We call this parameter the 〝min seg-size.〞 Fig. 10 shows an example of selection based on segment length. Increasing this value reduces the amount of data while improving the quality of the data.

IV. Experiments

Two experiments were conducted to evaluate the proposed method.

A. Experimental conditions

In this section, we show common experimental conditions. We used waveform data from 9 trees of songs A, B, ..., and I in nana′s database. The target post data was the full-length main melody sung by one female singer. The source post data were randomly selected from each tree, including posts singing backing chorus or partly singing.

Singing voice signals were sampled at 32 kHz, and acoustic features were extracted with a 5-ms shift. As an acoustic feature, 0^{th} through 43^{th} mel-cepstral coefficients were extracted from the smoothed spectrum analyzed by STRAIGHT[17]. The DNN used in this system was trained from mel-cepstral coefficients. The architecture of the DNN was a 3-hidden-layer feed-forward neural network with 1024 units per hidden layer. Acoustic features were normalized by the mean being zero and the variance being one. The mel-cepstral distortion between the target and the converted mel-cepstra was used as the objective evaluation measure, which is defined as:

$$\text{Mel-CD} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{D}(c_d^{(1)} - c_d^{(2)})^2}, \quad (1)$$

where $c_d^{(1)}$ and $c_d^{(2)}$ are the d^{th} coefficients of the target and the converted mel-cepstra, respectively.

B. Close test for parameter consideration

This experiment was carried out to determine the two parameters, the max distance (the maximum value of distance between matching path and diagonal) and the min seg-size (the minimum value of segment length). Combinations of the parameters were compared based on the mel-cepstral distortion. From the 9 trees, 8 trees (songs A, B, ..., and H) were selected. Then, 50 source posts and 1 target post were selected from each tree as the training data. The total training data of the source singers is 400 and the target singer is 8.

Table I and II show the experimental results. They describe Mel-CD and the percentage of extracted frames to all frames. In Table I, the best Mel-CD value in every column is bold-faced, and the best value in the table is underlined. In Table II, the corresponding cells are bold-faced and underlined. The value " - " means that no frames were extracted to train the model because no matching path satisfied the condition[s?]. Although more frames were used for training in the lower-left side of Table II, there are cells that have smaller Mel-CD values near the diagonal in Table I. This is because that there is a trade-off between the quantity and the quality of the extracted data. The quality of the data improved in the higher right side of the table where the values of max distance are smaller and the values of min seg-size are larger. These results indicate that our method with optimum parameters improves conversion accuracy.

C. Open test

In this experiment, four models were trained using the singing data of a different number of collaboration trees. They

TABLE I
Mel-CD [dB]

		Min seg-size (s)					
		0	0.1	1	3	5	7
Max distance (s)	0.025	5.925	5.993	-	-	-	-
	0.05	**5.922**	**5.921**	6.868	-	-	-
	0.1	5.924	5.931	6.063	6.930	-	-
	0.2	5.949	5.947	5.946	6.049	6.188	6.354
	0.3	5.979	5.973	**5.945**	5.959	6.0145	6.061
	0.4	5.991	5.990	5.972	**5.951**	5.965	5.981
	0.5	6.005	5.998	5.991	5.955	5.964	5.961
	0.6	6.013	6.010	6.003	5.963	**5.964**	5.968
	0.7	6.023	6.016	6.010	5.981	5.959	5.968
	0.8	6.035	6.026	6.014	5.993	5.975	**5.963**

Without the proposed method: 6.014 dB

TABLE II
The ratio of the number of frames used for training (%)

		Min seg-size [s]					
		0	0.1	1	3	5	7
Max distance (s)	0.025	10.12	2.67	-	-	-	-
	0.05	**18.08**	**16.70**	0.01	-	-	-
	0.1	29.91	29.39	2.78	0.01	-	-
	0.2	44.57	44.37	25.58	3.96	0.74	0.16
	0.3	53.73	53.61	**44.33**	19.66	8.42	3.81
	0.4	59.96	59.88	54.98	**36.88**	22.77	14.41
	0.5	64.52	64.45	63.18	48.85	35.94	26.49
	0.6	68.00	67.95	66.93	57.65	**46.69**	38.29
	0.7	70.78	70.74	69.91	63.28	55.03	47.42
	0.8	73.04	73.01	72.35	67.30	61.09	**54.76**

The total number of frames: 12,247,701

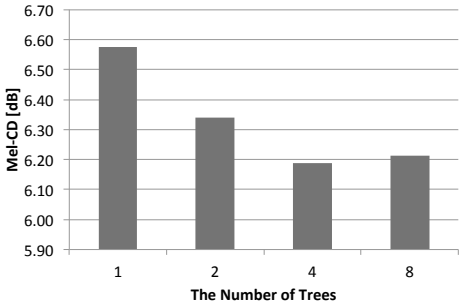

Fig. 11. Mel-CD [dB] (open test)

were compared for open data using the best parameters of our method in the previous experiment as follows:

- Max distance: 0.05 second
- Min seg-size: 0.1 second

The four models were trained using post data in 1 (song A), 2 (songs A and B), 4 (songs A, B, C, and D), and 8 (songs A, B, . . ., and H) trees, respectively. Four hundred post data sung by source singers were randomly selected from the tree/trees for all models as source singer training data. One post data was selected from each tree as a training data of the target singer. The test data set was composed of the post data in the tree of song "I," which included 18 source singers' data and one target singer's data.

Fig. 11 shows the results. Increasing the number of trees mostly caused a decrease in mel-cepstral distortion because the diversity of the training data improved. However, the model using 4 trees outperformed the model using 8 trees. It is assumed that each tree has different suitable parameters. Therefore, it is possible that parameters suitable for each tree are more largely differ in the model using 8trees than the model using 4 trees.

V. Conclusions

Using waveform data posted to social media, we proposed a method of extracting training data that can be used for many-to-one singing voice conversions. For training, we used the pairs of matched frames that have small differences in utterance timing. We assumed that posts that were sung with the same accompaniment would have little difference in utterance timing. Experimental results showed that setting two appropriate parameters (the maximum value of distance between the matching path and the diagonal and minimum value of segment length) while considering the trade-off between the quantity and the quality of the training data improved the objective evaluation measure. Increasing the number of trees used for training data caused us to accurately convert songs that were not used for training. Future works include researching proper parameters based on various elements (such as tempo), applying different parameters for each song, and subjective evaluation.

VI. Acknowledgment

This research was supported by nana music, Inc.

References

[1] J. Yin, W. Lo, and Z. Wu, "From Big Data to Great Services," 2016 IEEE International Congress on Big Data (BigData Congress), pp. 165–172, 2016.

[2] B. Li, T. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. Chin, K-C Sim, R. Weiss, K. Wilson, E. Variani, C. Kim, O. Siohan, M. Wein-traub, E. McDermott, R. Rose, and M. Shannon, "Acoustic Modeling for Google Home, ″ in INTERSPEECH-2017, Aug. 2017, pp. 399–403.

[3] P. Bell, M. J. F. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, and P. C. Woodland, "The MGB Challenge: Evaluating Multi-Genre Broadcast media recognition," IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[4] P. Karanasou, M. J. F. Gales, P. Lanchantin, X. Liu, Y. Qian, L. Wang, P. C. Woodland, and C. Zhang, "Speaker diarisation and longitudinal linking in multi-genre broadcast data," IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[5] J. Villalba, A. Ortega, A. Miguel, and L. Lleida, "Variational Bayesian PLDA for speaker diarization in the MGB Challenge," IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[6] P. C. Woodland, X. Liu, Y. Qian, C. Zhang, M. J. F. Gales, P. Karanasou, P. Lanchantin, and L. Wang, "Cambridge University transcription systems for the Multi-Genre Broadcast Challenge," IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[7] O. Saz, M. Doulaty, S. Deena, R. Milner, R. Ng, M. Hasan, Y. Liu, and T. Hain, "The 2015 Sheffield system for transcription of multi-genre broadcast media," IEEE Automatic Speech Recognition and Understanding Workshop, 2015.

[8] nana, https://nana-music.com/ (2018)

[9] T. Toda, A. W. Black, and K. Tokuda, "Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter," ICASSP 2005, 2005

[10] 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, 2007.

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

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

[13] N. Hosaka, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Voice Conversion Based on Trajectory Model Training of Neural Networks Considering Global Variance," Interspeech 2016, 2016.

[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, No. 1, pp. 43–49, 1978.

[15] Y. Hono, K. Sawada, K. Hashimoto, K. Oura, Y. Nankaku, K. Tokuda, D. Kondo, and D. Ishikawa "Singing voice conversion using post data in music SNS," Proc. of Acoustical Society of Japan Autumn Meeting, 1-8-16, pp. 209–210, 2017 (in Japanese).

[16] K. Yoshida and H. Sakoe, "Online Handwritten Character Recognition for a Personal Computer System," IEEE Transactions on Consumer Electronics, vol. CE-28, No. 3, pp. 202–209, 1982.

[17] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," Speech Communication, Vol. 27, No. 3–4, pp. 187–207, 1999.