Odyssey 2018 The Speaker and Language Recognition Workshop
26-29 June 2018, Les Sables d'Olonne, France

sprocket: Open-Source Voice Conversion Software

Kazuhiro Kobayashi, Tomoki Toda

Information Technology Center, Nagoya University, Japan

kobayashi.kazuhiro@g.sp.m.is.nagoya-u.ac.jp, tomoki@icts.nagoya-u.ac.jp

Abstract

Statistical voice conversion (VC) is a technique to convert specific non- or paralinguistic information while keeping linguistic information unchanged, and speaker conversion has been studied as a typical application of VC for a few decades. To better understand various VC techniques using a freely available common dataset, the Voice Conversion Challenge (VCC) was launched in 2016 and the 2nd challenge was held in 2018. As one of the baseline systems for VCC 2018, we developed open-source VC software called "sprocket", in which not only conventional techniques, such as a trajectory-based conversion method using a Gaussian mixture model (GMM) and a vocoder-based conversion framework but also recently developed techniques, such as a vocoder-free VC framework, have been implemented. Using sprocket, it is possible to 1) easily reproduce converted voices using the VCC datasets and 2) develop VC systems using other parallel speech datasets with fundamental VC functions, such as acoustic feature extraction, time alignment between the source and target features, GMM training, feature conversion, and waveform generation. In this paper, we describe 1) the technical details and use of sprocket, 2) the development of the baseline systems for the HUB and SPOKE tasks of VCC 2018 using sprocket, and 3) the performance of sprocket as a VC system by demonstrating results for our developed baseline systems in VCC 2018.

1. Introduction

Variations of voice characteristics, such as fundamental frequency (F_0) patterns and voice timbre characteristics produced by individual speakers are usually restricted by their physical constraints due to the speech production mechanisms. These constraints based on the speech production mechanisms are helpful for producing a speech signal conveying not only linguistic information for communication but also paralinguistic information such as speaker individuality and emotions. However, they sometime generate various barriers to producing desired voice characteristics, such as desired speech expressions and voice quality. If individual speakers could freely produce various voice characteristics without being limited by their physical constraints, it would break down these barriers and open up an entirely new speech communication style.

Voice conversion (VC) is one of the potential techniques enabling speakers to produce speech sounds beyond their physical constraints [1]. VC research was originally started to develop a speaker individuality conversion technique enabling a source speaker to change his/her speaker individuality to that of another target speaker while preserving the linguistic content [2]. The conversion frameworks of VC have been adopted into other research objectives such as speech recovery for people with speech disorders [3], singing style conversion [4], nonnative to native speaker conversion [5], and speech to articulatory mapping [6] to make it possible to implement augmented speech communications. Towards the practical use of these VC applications, it is essential to improve fundamental VC techniques.

In this study, we focus on the conversion of speaker individuality. To convert a source speaker individuality into a target speaker individuality, conversion functions are trained using a set of speech utterances of the source and target speakers. In VC research, it is usually assumed that the set consists of the same linguistic contents for the source and target speakers (i.e., a parallel dataset) to model the conversion functions. For modeling using parallel dataset, several techniques such as the Gaussian mixture model (GMM) [7, 8, 9], Gaussian process regression [10, 11], non-negative matrix factorization [12, 13], and deep neural networks [14, 15, 16] have been proposed. To make it possible to implement VC when the set consists of different linguistic contents (i.e., a nonparallel dataset), several nonparallel VC techniques such as feature alignment [17, 18], speaker adaptation [19, 20, 21], and direct modeling [22, 23] have been proposed. Although these techniques make it possible to convert the speaker individuality using either a parallel or non-parallel dataset, the sound quality of the converted voice and the conversion accuracy of the speaker individuality are usually degraded compared with those of the target voice.

To better understand various VC techniques using a freely available common dataset, the Voice Conversion Challenge (VCC) was launched in 2016, and the 2nd challenge was held in 2018 [24]. As one of the baseline systems for VCC 2018, we developed open-source VC software called "sprocket", in which not only conventional techniques, such as a trajectory-based conversion method with a GMM [8] and a vocoder-based framework, but also recently developed techniques, such as a vocoder-free VC framework based on a differential GMM (DIFFGMM) [25, 26] and an F_0 transformation technique using waveform modification [27], have been implemented. Using sprocket, it is possible to 1) easily reproduce the converted voices using the VCC datasets and 2) develop VC systems using other parallel datasets with fundamental VC functions, such as acoustic feature extraction, time alignment between the source and target features, GMM training, feature conversion, and waveform generation. In this paper, we describe 1) the technical details and use of sprocket, 2) the development of the baseline systems for the HUB and SPOKE tasks of VCC 2018 using sprocket, and 3) the performance of sprocket as a VC system by demonstrating results for our developed baseline systems in VCC 2018.

The rest of this paper is organized as follows. The GMM-based VC techniques implemented in sprocket are described in Section 2. The use of sprocket is described in Section 3. The system settings for VCC 2018 are described in Section 4. In Section 5, we describe the performance of sprocket as the baseline system in VCC 2018. Finally, the conclusion is given in

10.21437/Odyssey.2018-29

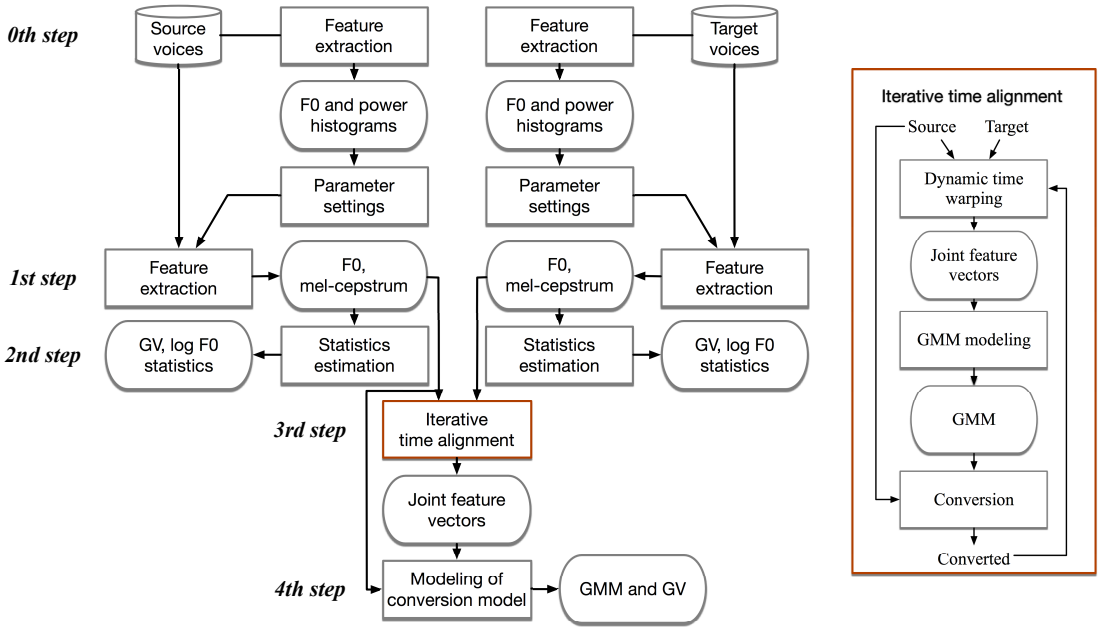

Figure 1: Training process of the GMM-based VC method using a parallel dataset.

Section 6.

2. GMM-based VC methods in sprocket

Statistical VC is a technique to convert the individuality of a source speaker into that of a target speaker by converting several acoustic features such as F_0, aperiodicity, and the mel-cepstrum. To convert these features, VC usually requires two steps, i.e., training and conversion processes. In this section, we describe details of a GMM-based VC method using parallel speech utterances of the source and target speakers (i.e., a parallel dataset), focusing on two typical methods: 1) maximum likelihood parameter generation (MLPG) considering the global variance (GV) based on the GMM [8], 2) vocoder-free VC using the log-spectral differential (DIFFVC) [25, 26], which have been implemented in sprocket.

The GMM-based VC method includes the following techniques: 1) MLPG considering the GV based on the GMM, 2) vocoder-free VC using the DIFFVC. These processes have been implemented in the open-source VC software "sprocket", whose use is described in Section 3.

2.1. Training process

Figure 1 shows the training process of the GMM-based VC method. For the training process, the GMM-based VC method carries on following steps: 0) preparation of the parallel speech dataset, 1) acoustic feature extraction, 2) calculation of acoustic feature statistics, 3) time alignment between the source and target feature vectors, and 4) GMM modeling.

Preparation (0th step): To train a conversion model, it is necessary to prepare parallel speech utterances consisting of the same linguistic information (i.e., the same phonemes, syllables, and words) and different speaker individualities. To prepare such a dataset, speech utterances uttered by the source and target speakers using the same manuscripts (e.g., 50 uttered sentences, each with a duration of about 3 or 5 seconds) are usually used.

1st step: Acoustic features, including F_0, aperiodicity, and mel-cesptrum, parameterized from the spectral envelope are extracted from the speech signals of both the source and target speakers. Because the error in the acoustic feature extraction always propagates to the subsequent steps and strongly affects the resulting quality of statistical VC, it is very important to carefully set the configuration parameters for the acoustic feature extraction process (e.g., the range of F_0 extraction).

2nd step: In this step, speaker-dependent statistics of the extracted acoustic features, such as the mean and standard deviation of the logarithmic F_0 and the GV of the mel-cepstrum, are estimated.

3rd step: To model a joint probability density function based on the GMM, frame-aligned joint feature vectors are required. However, the speech signals of the source and target speakers are not usually aligned because these speakers usually utter with different speaking styles even when using the same manuscripts. To align the source and target feature vectors frame by frame, an iterative alignment process based on dynamic time warping (DTW) is performed as follows:

A. To remove silent parts of the feature vectors, frame-based power thresholding is performed after removing the zeroth coefficient of the mel-cepstrum. Then, the resulting static feature vectors are extended to static and delta feature vectors.

B. To estimate time-warping functions between the source and target feature vectors in each utterance, DTW is performed to minimize a distance metric between the aligned source and target feature vectors. For the mel-cepstrum, mel-cepstrum distortion is usually used as the distance metric.

C. Applying the estimated time-warping functions to the source and target feature vectors, the frame-aligned joint feature vectors are constructed.

D. A joint probability density function based on the GMM

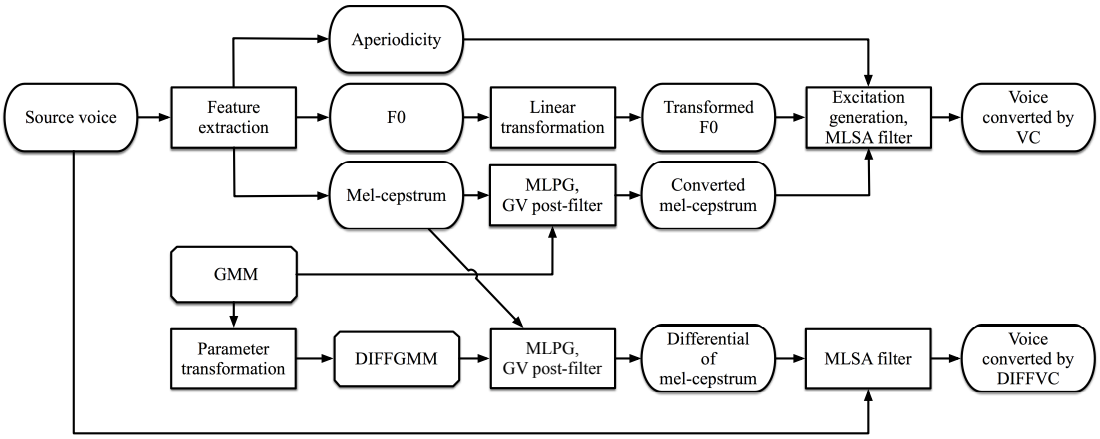

Figure 2: Conversion process of the VC and DIFFVC methods (**5th step**).

is trained based on the expectation-maximization algorithm using the joint feature vectors.

E. The static and delta feature vectors of the source with silent parts are converted into static feature vectors of the target speakers by MLPG using the GMM.

F. Step **A** is applied to the converted feature vectors.

G. By applying step **B** to the converted feature vectors and target feature vectors, the time-warping functions are refined because the converted feature vectors have the same temporal structure as the source feature vectors and similar speaker individuality to the target speaker.

H. The process returns to step **C** until the final iteration.

This iterative time alignment processing is performed to constructing joint feature vectors of the mel-cepstrum. Those of the other acoustic features such as aperiodicity are usually constructed using the resulting time-warping function of the mel-cepstrum.

4th step: Using the iteratively refined joint feature vectors, the joint probability density function based on the GMM is trained as the conversion model for the conversion process. The GV of the converted feature vectors is also calculated to design a GV post-filter.

2.2. Conversion process (5th step)

For the conversion process, the acoustic features of the source speaker are converted into those of the target speaker using the trained GMM. As the acoustic feature to be converted in sprocket, F_0 and the mel-cepstrum are used. Other factors such as the aperiodicity, speaking rate, the temporal structure of the F_0 trajectory, and the power trajectory are retained as those of the source voice. Note that arbitrary utterances of the source speaker can be converted into those of the target speaker. Figure 2 shows the conversion process of the VC based on the GMM and the DIFFVC based on a differential GMM (DIFFGMM).

First, F_0, the mel-cepstrum, and aperiodicity are extracted from a source voice. For the GMM-based VC method [28], F_0 is linearly transformed frame by frame using the speaker-dependent statistics of the source and target speakers in the logarithmic space. The mel-cepstrum is converted into that of the target speaker by MLPG [28] after constructing the static and delta feature vectors without zeroth order of the mel-cepstrum coefficients. Then, the GV post-filter is applied to the converted mel-cepstrum because the GV of the converted mel-cepstrum is usually degraded compared with that of the original mel-cepstrum, and the zeroth order of the source mel-cepstrum is concatenated to the resulting converted mel-cepstrum. Here, to ensure the same waveform power for the source and converted voices, the zeroth order of the converted mel-cepstrum is modified. Finally, the voice converted by the VC method is generated by using excitation generation and the mel log spectral approximation (MLSA) filter [29] (i.e., a vocoder) using the transformed F_0 and converted mel-cepstrum.

For the DIFFVC based on the DIFFGMM [25, 26], the model parameters of the trained GMM are modified from the joint probability density of the source and target features to a joint probability density of the source feature and a feature differential between the target and source features. Then, the converted mel-cepstrum differential is estimated from the source mel-cepstrum by MLPG with the DIFFGMM. The voice converted by the DIFFVC method is generated by filtering the source voice using the GV post-filtered mel-cepstrum differential, where the MLSA filter is also used. Although the DIFFVC based on the DIFFGMM makes it possible to achieve converted voice with significantly higher sound quality than that obtained by VC based on the GMM method, the conversion accuracy of the speaker similarity significantly decreases when performing the conversion for speakers with different gender (i.e., cross-gender VC) because there is no F0 transformation module when using the vocoder.

2.3. DIFFVC with F_0 transformation

By applying the vocoder-free F_0 transformation to waveform signals of the source speaker, it is possible to take advantage of the vocoder-free framework for not only same-gender conversion but also cross-gender conversion by the DIFFVC method [27].

First, the F_0 transformation ratio is calculated from the mean values of F_0 for the source and target speakers. Then, the waveforms of the source speaker are transformed in accordance with the F_0 transformation ratio using duration modification techniques and resampling. For the F_0 transformation technique, the waveform similarity-based overlap and add (WSOLA) method [30] has been implemented in sprocket. Note that this F_0 transformation process changes the voice

timbre of the source voice. By performing the training and conversion processes in the same manner as described in Sections 2.1 and 2.2 using the F_0-transformed source voices as the source voices, the DIFFVC method with the transformation is achieved [27].

3. Use of sprocket

sprocket [31] is open-source software that converts speaker individuality using the GMM-based VC methods with a parallel dataset. sprocket aims to provide an environment for both expert and non-expert users to easily use the statistical VC framework. The license of sprocket is set to the MIT license [32] so that its features can be freely applied not only for research purposes but also for industrial purposes. In this section, we describe how to use sprocket.

3.1. Installation

Figure 3 shows the directory structure of sprocket. sprocket is open to the public on a GitHub repository. Python3 is adopted as the main programming language and we assume that users use sprocket on a Unix environment. First, users need to install the dependent libraries via the pip command. Then, by executing `python3 setup.py install` in a terminal, the libraries using sprocket are installed in the Python3 environment. More details of the installation are given on the top page of the GitHub repository.

3.2. Preparation of speech dataset and configure files

In this subsection, we assume that the working directory is set to `example/`.

Preparation of speech dataset

For statistical VC, it is necessary to prepare a parallel dataset consisting of the same speech utterances uttered by the different source and target speakers. To execute sprocket without preparing a speech dataset, we have prepared a script to automatically download the speech dataset of VCC 2016 [33] and deploy it in the correct place. To use arbitrary speech datasets, it is better to prepare a speech dataset more than 50 utterances of the source and target speakers because the conversion accuracy strongly depends on the number of training utterances. In Section 3.5, we describe the preparation of arbitrary datasets in more detail.

In sprocket, the supported file format of the speech signals is 16000 Hz, 22050 Hz, 44100 Hz, or 48000 Hz for the sampling rate, single channel, and 16-bit signed-integer waveform. The waveforms are stored in not a single waveform file but several waveform files by dividing into several utterances of about 5 seconds each. These waveform files must be deployed in `data/wav` (e.g., `data/wav/speakerA/*.wav` for speakerA) for each speaker.

Initialization

To generate list files and speaker-dependent and pair-dependent configure files for use in the training and conversion processes, `initialize.py` is executed. `initialize.py` takes three arguments. The first argument is for the source speaker label (e.g., speakerA), the second argument is for the target speaker label (e.g., speakerB), and the third argument is for the sampling rate of the format (e.g., 16000).

The lists showing paths of the waveform files (e.g., speakerA_train.list for the training and speakerA_eval.list for the evaluation), the speaker-dependent YAML files (e.g., speakerA.yml) showing the format of the waveform files and parameters for acoustic feature extraction, and the pair-dependent

```
sprocket # project home
├── example # framework
│   ├── conf
│   │   ├── default
│   │   ├── figure
│   │   ├── pair
│   │   └── speaker
│   ├── data
│   │   ├── pair
│   │   └── wav
│   ├── initialize.py
│   ├── list
│   ├── run_f0_transformation.py
│   ├── run_sprocket.py
│   └── src
├── requirements.txt
├── setup.py
└── sprocket # library
```

Figure 3: Directory structure of sprocket.

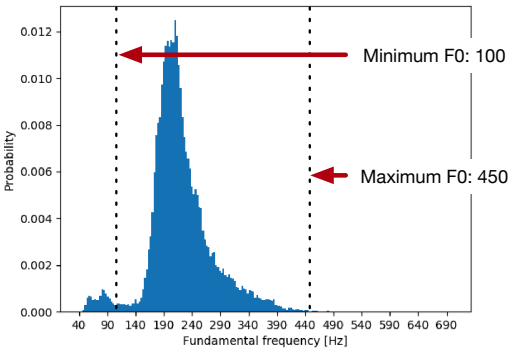

Figure 4: Example of the F_0 range setting of a female speaker (VCC2SF1 in VCC 2018 dataset).

YAML (e.g., speakerA-speakerB.yml) file showing the parameters used in the GMM modeling are generated in the corresponding directories such as `list`, `conf/speaker`, and `conf/pair`, respectively. Also, the histograms for the settings of the F_0 range and power thresholding are generated in the `conf/figure` directory.

Modification of the lists

It is necessary to modify the automatically generated lists to select the training and evaluation utterances. The training lists are used for the calculation of the speaker-dependent statistics and the GMM modeling. The evaluation lists describe the paths of the waveform during the evaluation. The initially generated lists contain the paths of all waveform files deployed in `data/wav` for each speaker. Because the waveforms used for the training and evaluation should be independent, the users should erase the overlapping paths in the training and evaluation files. Note that the order of the listed waveform files should be the same for the source and target speakers.

Setting of the speaker-dependent parameters

The F_0 range is a representative speaker-dependent parameter for acoustic feature extraction. For normal speech, the F_0 range is relatively well approximated as a unimodal distribution. In the F_0 extraction process, F_0 values of the double and half harmonics are sometimes extracted owing to the analysis errors, and these errors significantly degrade the sound quality of the converted voice. To avoid such errors, the F_0 range for each

speaker is specified in accordance with the F_0 histograms.

To decide the F_0 range, F_0 histograms are generated using the results of F_0 extraction without specifying any F_0 range by executing `initialize.py`. Figure 4 shows an example of F_0 range setting using an F_0 histogram. Using the histograms, the F_0 range in the speaker-dependent YAML file is modified. The maximum and minimum values of the F_0 range are decided from the left and right tails by considering the histogram as a unimodal distribution, respectively.

Setting of the pair-dependent parameters
The pair-dependent YAML file is generated in the `conf/pair` directory by executing `initialize.py`. Because the number of mixture components of the GMM strongly affects the conversion quality, it should be carefully set in accordance with the number of training utterances. As a guideline, the number of mixture components is set to 8 or 16 when the number of training utterances is 30 and the number of mixture components is set to 32 when the number of training utterances is 50 where it is assumed that each utterance consists of a speech signal of about 5 seconds. If users train the GMM with more training utterances such as 150 utterances, it is appropriate to set the number of mixture components to 32 or 64.

The other pair-dependent parameter is the coefficient for the GV post-filter. The GV post-filter is used to increase the variance of the converted feature trajectories, which improves the sound quality of the converted voice. However, it sometimes causes serious sound quality degradation. If the converted voice severely suffers from artifact sounds, it is worth decreasing the GV post-filter coefficient from 1.0.

3.3. Training and conversion processes

Users can execute statistical VC if the parameters described in the speaker-dependent and pair-dependent YAML files have been set correctly. The main script `run_sprocket.py` takes teh two arguments of the source and target speaker labels and some options corresponding to the following steps (e.g., `python3 run_sprocket.py -1 -2 -3 -4 -5 speakerA speakerB`). Note that all files generated by sprocket are stored below the pair-dependent directory (e.g., `data/pair/speakerA-speakerB/`).

Step 1: In sprocket, WORLD [34] is used for speech parameter extraction and vocoder synthesis because it achieves high sound quality with a low computational cost. Acoustic features such as F_0, the mel-cepstrum, and aperiodicity are extracted. These acoustic features are stored as files in HDF5 file format for each utterance. Because WORLD was developed using C++, sprocket uses a wrapper framework called "PyWorldVocoder" [35].

Step 2: In this step, the speaker-dependent statistics such as the mean and standard deviation of F_0 and the GV of the mel-cepstrum are calculated. In this process, the statistics are stored in the `stats` directory.

Step 3: In this step, to construct joint feature vectors for the GMM modeling, the iterative time-alignment estimation is performed. As described in Section 2, the iterative process consists of the DTW, GMM modeling, and conversion of the source feature vectors. The number of the iterations is given in the pair-dependent YAML file. The resulting joint feature vectors are stored in the `jnt` directory.

Step 4: The GMM for the mel-cepstrum is trained. Also, the GV of the converted feature vector is calculated using the trained GMM. These parameters are stored in the `model` directory.

Step 5: The source voices described in the source speaker evaluation list file are converted into target voices. The voices converted by VC are labeled as `*_VC.wav` and those converted by the DIFFVC are labeled as `*_DIFFVC.wav`. These converted voices are saved in the `test` directory. Note that by adding an option "-5" when executing sprocket (e.g., `python3 run_sprocket.py -5 speakerA speakerB`), it is possible to perform only the conversion process.

3.4. F_0 transformation based on waveform modification

It is possible to transform F_0 for the source voices using `run_f0_transformation.py`. This script employs two arguments in the same manner as `run_sprocket.py`. Using this script, the F_0-transformed source voices based on the duration modification and resampling techniques are generated in the `data/wav` directory. The F_0-transformed waveforms are named as the source speaker with the F_0 transformation ratio (e.g., speakerA_1.45). By executing initialization, training, and conversion using the F_0-transformed source voices and target voices, the users can achieve the DIFFVC with the F_0 transformation.

3.5. Tips on system development

It is possible to convert speaker individuality within an arbitrary speaker pair by using sprocket if users prepare a parallel speech dataset. To obtain a high-quality converted voice, it is essential to prepare a well-constructed speech dataset. In this section, we give some tips for preparing such a speech dataset.

For the speech waveforms of the target speaker, it is desirable that the waveforms are recorded in a high-quality sound environment because the sound quality of the target voices strongly affects the quality of the converted voices. If users employ low-quality target waveforms recorded under noisy or reverberant conditions, the sound quality of the converted voice will be significantly degraded.

Regarding the sound quality of the source voices, it is recommended that the source voices are recorded in the same environment as that used in the VC system, i.e., the training and evaluation utterances should be recorded in the same environment. Also, because sprocket does not convert speaking styles such as F_0 patterns or the speaking rate, it is recommended that the source speaker tries to imitate the target speaker's speaking style.

If users implement a male-to-female speaker conversion, it is acceptable to record the source voice uttered in a falsetto to match the value of F_0 in order to perform the DIFFVC method without F_0 transformation. To develop appropriate joint feature vectors, it is better to control short pause positions so that they correspond to each other for the source and target speakers. It will be convenient to decide the pause positions in each utterance in accordance with the target voices when recording the source voices.

4. Development of VCC 2018 baseline systems with sprocket

VCC 2018 was a challenge in which several VC techniques are evaluated using same speech datasets provided by the organizers. The challenge compared VC systems submitted by teams from universities, research institutes, and industry in terms of the sound quality and conversion accuracy of the speaker in-

dividuality. In VCC 2018, there were two tasks called the HUB task and SPOKE task. For the HUB task, parallel speech datasets with two male and two female source and target speakers were provided. For the SPOKE task, non-parallel speech datasets consisting of two male and two female source speakers and two male and two female target speakers were provided. In this section, we describe the development of the VCC 2018 baseline system for each task using sprocket.

4.1. Baseline system for the HUB task

As described in Section 3, sprocket can carry out VC using GMM-based VC methods including VC and DIFFVC with a parallel speech dataset. In terms of the sound quality, the DIFFVC method is usually superior to the VC method for same-gender speaker pairs owing to the vocoder-free framework. On the other hand, for cross-gender speaker pairs, there is no significant difference between VC and DIFFVC employing F_0 transformation methods. For the conversion accuracy of the speaker individuality, there is no significant difference between the VC and DIFFVC methods. To maximize the sound quality of the converted voice, for the HUB task, we chose the DIFFVC method without F_0 transformation for the same-gender speaker pairs and the VC method for the cross-gender speaker pairs.

4.2. Baseline system for the SPOKE task

It is not possible to model any GMM using the non-parallel speech datasets because time-alignment using non-parallel speech datasets cannot be dealt with using sprocket. Fortunately, parallel datasets between the source speakers and target speakers using in the evaluation were provided in the HUB task. To build the conversion function for the SPOKE task, we modeled gender-dependent speaker-independent GMMs using the dataset provided for the HUB task. Because the speaker-independent GMMs were mismatched models for the used for evaluation in the SPOKE task, it was expected that the sound quality and conversion accuracy of the speaker individuality would significantly decrease compared with those for matched models. For the SPOKE task, we submit voices converted by DIFFVC for the same-gender speaker pairs and those based on the VC for the cross-gender speaker pairs.

5. Results of VCC 2018

5.1. Baseline system conditions

We used the English speech database provided by VCC 2018. For the HUB task, the number of source speakers was four, two females and two males, and the number of target speakers was four, two females and two males who were different from the source female and male speakers. For the SPOKE task, four additional source speakers, two females and two males and target speakers for the HUB tasks were used. The number of sentences uttered by each speaker was 116, 81 utterances for the training and 35 utterances for the evaluation. The sampling frequency was set to 22050 Hz.

WORLD [34] was used to extract spectral envelopes, which were parameterized into 1-35 mel-cepstral coefficients as the spectral feature. The frame shift was 5 ms. The mel log spectrum approximation (MLSA) filter [36] was used as the synthesis filter. As the source excitation features, we used F_0 and aperiodicity extracted using WORLD [34]. Table 1 shows the speaker-dependent parameters for the VCC 2018 datasets.

For the HUB task, the speaker-dependent GMMs were separately trained for all combinations of the source and target speakers. The full covariance was used for the GMMs. The number of mixture components of the GMMs was 32. For the SPOKE task, gender-dependent speaker-independent GMMs were separately trained for each target speaker using speech datasets consisting of the source speakers and target speaker used in the HUB task.

Table 1: Speaker-dependent parameters for the VCC 2018 datasets. Note that default indicates a value unchanged from the default setting.

Speaker	Minimum F_0 [Hz]	Maximum F_0 [Hz]	Power [dB]
VCC2SF1	100	450	-31
VCC2SF2	110	350	-31
VCC2SF3	130	330	default
VCC2SF4	120	390	default
VCC2SM1	50	200	-31
VCC2SM2	70	300	-40
VCC2SM3	60	240	default
VCC2SM4	60	270	default
VCC2TF1	140	350	-45
VCC2TF2	100	400	-30
VCC2TM1	60	200	-23
VCC2TM2	50	280	-31

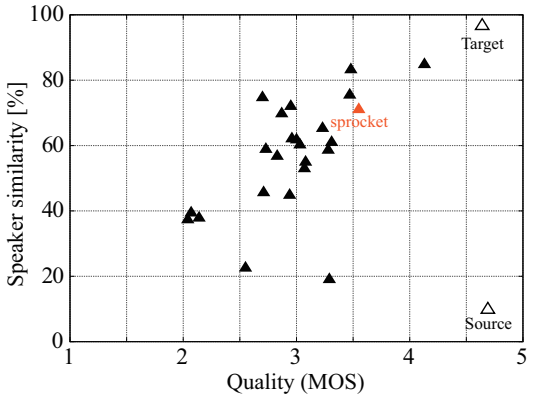

Figure 5: Overall results for the HUB task.

5.2. Results for the HUB task

Figure 5 shows the overall results for the HUB task. The baseline VC system achieved a reasonably high sound quality with a mean opinion score (MOS) of over 3.5 and over 70% similarity for all the submitted systems.

Figures 6 and 7 show MOSs for the sound quality for the same-gender and cross-gender speaker pairs, respectively. In Figure 6, the baseline system (B01) achieves the second highest sound quality for the same-gender speaker pairs. This shows that the DIFFVC method makes it possible to achieve very high sound quality owing to the vocoder-free framework, which was only exceeded by the N10 system. As shown in Figure 7, the baseline system (B01) achieved seventh place for the cross-gender speaker pairs. By comparing the sound quality results of the baseline system for the same-gender and cross-gender speaker pairs, the MOS for the sound quality decreases by about 1.0 owing to the use of the vocoder to generate the F_0-transformed excitation signal.

Figure 8 shows results for the speaker similarity of the converted voice. Here, the baseline system (B01) achieves rela-

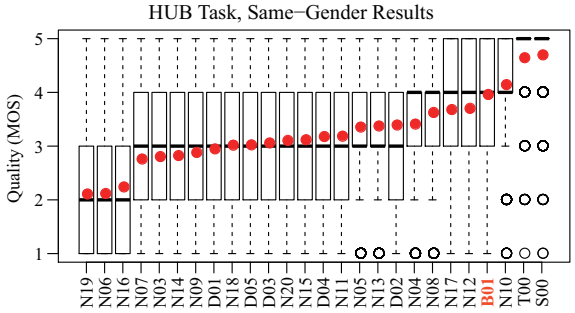

Figure 6: Results for sound quality for same-gender speaker pairs.

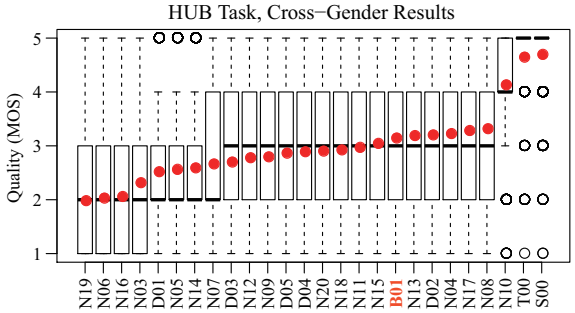

Figure 7: Results for sound quality for cross-gender speaker pairs.

tively good performance compared with the other systems even though it does not use nonlinear conversion functions in contrast to Gaussian process and deep neural networks, but mixtures of linear functions based on the GMM. It is possible that the iterative time alignment process generates better joint feature vectors, resulting in more sophisticated conversion models.

5.3. Results on the SPOKE task

Figure 9 shows the overall results for the SPOKE task of VCC 2018. The sound quality and conversion accuracy of the speaker individuality are significantly lower than those for the HUB task because the conversion models used mismatched models for the source speakers in the SPOKE task. The degradation using the mismatched models was about 1.0 of the MOS for the sound quality and the degradation of the conversion accuracy was about 15% compare with those for the HUB task.

6. Conclusion

In this paper, we have described in detail the baseline system based on "sprocket" used for the Voice Conversion Challenge (VCC) 2018. The baseline system consists of statistical voice conversion (VC) techniques based on a Gaussian mixture model (GMM) and a vocoder-free VC technique based on a differential GMM (DIFFVC). For the HUB task in VCC 2018, the baseline system achieved second for sound quality owing to the vocoder-free VC framework for the same-gender speaker pairs and seventh place for the cross-gender speaker pairs. In the evaluation of the speaker similarity, the baseline system achieved the sixth place among the submitted systems. For the SPOKE task, the baseline system achieved an average position even though it used mismatched conversion models trained using different source speakers. In this paper, we also described sprocket in detail as well as its uses, will be useful for the VC research community up. In future work, we will attempt to improve the usability of sprocket and add some conversion techniques based on deep learning.

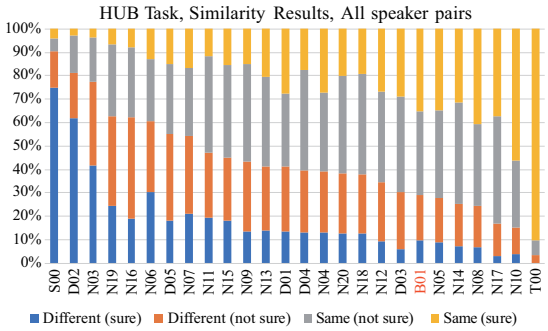

Figure 8: Results for speaker similarity.

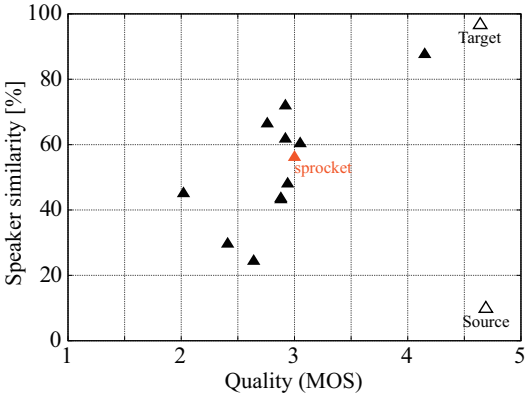

Figure 9: Overall results for the SPOKE task.

7. Acknowledgements

The authors would like to thank Ryuichi Yamamoto and Tatsunori Uchino for technical assistance with sprocket implementation. This work was partly supported in part by JSPS KAKENHI Grant-in-Aid for JSPS Research Fellow Number 16J10726, and by JST, PRESTO Grant Number JPMJPR1657.

8. References

[1] T. Toda, "Augmented speech production based on real-time statistical voice conversion," *Proc. GlobalSIP*, pp. 755–759, Dec. 2014.

[2] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *J. Acoust. Soc. Jpn (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[3] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "An evaluation of alaryngeal speech enhancement methods based on voice conversion techniques," *Proc. ICASSP*, pp. 5136–5139, May 2011.

[4] K. Kobayashi, T. Toda, H. Doi, T. Nakano, M. Goto, G. Neubig, S. Sakti, and S. Nakamura, "Voice timbre control based on perceived age in singing voice conversion," *IEICE Trans. Inf. Syst.*, vol. E97-D, no. 6, pp. 1419–1428, 2014.

[5] S. Aryal and R. G. Osuna, "Can voice conversion be used to reduce non-native accents?," *Proc. ICASSP*, pp. 7929–7933, May 2014.

[6] P. L. Tobing, K. Kobayashi, and T. Toda, "Articulatory controllable speech modification based on statistical inversion and production mappings," *IEEE/ACM Trans. ASLP*, vol. 25, no. 12, pp. 2337–2350, Dec. 2017.

[7] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. SAP*, vol. 6, no. 2, pp. 131–142, Mar. 1998.

[8] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Trans. ASLP*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[9] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," *Proc. INTERSPEECH*, Sept. 2012.

[10] N. Pilkington, H. Zen, and M. Gales, "Gaussian process experts for voice conversion," *Proc. INTERSPEECH*, pp. 2761–2764, Aug. 2011.

[11] N. Xu, Y. Tang, J. Bao, A. Jiang, X. Liu, and Z. Yang, "Voice conversion based on Gaussian processes by coherent and asymmetric training with limited training data," *Speech Communication*, vol. 58, pp. 124–138, Mar. 2014.

[12] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Trans. Inf. and Syst.*, vol. E96-A, no. 10, pp. 1946–1953, Oct. 2013.

[13] Z. Wu, T. Virtanen, E. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. ASLP*, vol. 22, no. 10, pp. 1506–1521, June 2014.

[14] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," *Proc. INTERSPEECH*, pp. 369–372, Aug. 2013.

[15] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Trans. ASLP*, vol. 22, no. 12, pp. 1859–1872, Dec. 2014.

[16] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," *Proc. ICASSP*, pp. 4869–4873, Apr. 2015.

[17] D. Erro, A. Moreno, and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE Trans. ASLP*, vol. 18, no. 5, pp. 944–953, 2010.

[18] T. Hashimoto, D. Saito, and N. Minematsu, "Arbitrary speaker conversion based on speaker space bases constructed by deep neural networks," *Proc. APSIPA*, pp. 1–4, Dec. 2016.

[19] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Many-to-many eigenvoice conversion with reference voice," *Proc. INTERSPEECH*, pp. 1623–1626, Sept. 2009.

[20] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice conversion with smoothed GMM and MAP adaptation," *Proc. INTERSPEECH*, pp. 1–4, Sept. 2003.

[21] C.-H. Lee and C.-H. Wu, "Map-based adaptation for speech conversion using adaptation data selection and non-parallel training," *Proc. INTERSPEECH*, pp. 17–21, Sept. 2006.

[22] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," *Proc. APSIPA*, pp. 1–6, Dec. 2016.

[23] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," *Proc. INTERSPEECH*, Apr. 2017.

[24] J. L. Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *Proc. Odyssey*, June 2018 (Submitted).

[25] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," *Proc. INTERSPEECH*, pp. 2514–2518, Sept. 2014.

[26] K. Kobayashi, T. Toda, and S. Nakamura, "Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential," *Speech Communication*, Mar. 2018 (In press).

[27] K. Kobayashi, T. Toda, and S. Nakamura, "F0 transformation techniques for statistical voice conversion with direct waveform modification with spectral differential," *Proc. IEEE SLT*, pp. 693–700, Dec. 2016.

[28] T. Toda, Y. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," *Proc. ICASSP*, pp. 1249–1252, Apr. 2007.

[29] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (MLSA) filter for speech synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[30] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," *Proc. ICASSP*, vol. 2, pp. 554–557, Apr. 1993.

[31] k2kobayashi, "sprocket," `https://github.com/k2kobayashi/sprocket`.

[32] "The MIT License," `https://opensource.org/licenses/MIT`.

[33] T. Toda, L. Chen, S. Daisuke, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," `http://datashare.is.ed.ac.uk/handle/10283/2042`.

[34] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. on Info. and Syst.*, vol. E99-D, no. 7, pp. 1877–1884, 2016.

[35] JeremyCCHsu, "Python-wrapper-for-world-vocoder," `https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder`.

[36] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis – a unified approach to speech spectral estimation," *Proc. ICSLP*, pp. 1043–1045, 1994.