Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

Personalized Singing Voice Generation Using WaveRNN

Xiaoxue Gao, Xiaohai Tian, Yi Zhou, Rohan Kumar Das and Haizhou Li

Department of Electrical and Computer Engineering
National University of Singapore, Singapore
{xiaoxue.gao, yi.zhou}@u.nus.edu, {eletia, rohankd, haizhou.li}@nus.edu.sg

Abstract

In this paper, we formulate a personalized singing voice generation (SVG) framework using WaveRNN with non-parallel training data. We develop an average singing voice generation model using WaveRNN from multi-singer's vocals. To map singing Phonetic PosteriorGrams and prosody features from singing template to time-domain singing samples, a *speaker i-vector* extracted from target speech is used to control the speaker identity of the generated singing. At run-time, a singing template and target speech samples are used for target singing vocal generation. Specifically, the content and the speaker identity of the target speech is not necessarily the same as that of the singing template. Experimental results on the NUS-48E and NUS-HLT-SLS corpora suggest that the personalized SVG framework outperforms the traditional conversion-vocoder pipeline in the subjective and objective evaluations.

1. Introduction

Personalized singing voice generation, that we call SVG for brevity, aims to generate singing for a user according to a standard singing template and the user's speaking voice. The SVG task is challenging, because we expect the generated singing to, a) be as natural as the standard singing template, where the template is recorded from the trained singers, b) follow the same tempo or rhythm of the template, and c) be similar to the source (user's) voice identity, which is different from the template singer.

The prior works have addressed some of the above challenges. For example, template-based STS conversion [1, 2] and SVC [3] techniques suggest an analysis-modification-synthesis pipeline as the solution. Another technique along this direction, singing synthesis [4–7], follows the idea of speech synthesis, which considers text and sheet music as input and generates high-quality singing as the output [8–11].

The template-based STS conversion approach hinges on the quality of frame alignment between input spoken lyrics and the singing template [1, 2, 12, 13], and appropriate spectral and prosody mapping [14–16]. Due to the significant differences between the input speech and the template singing, the accuracy of alignment and the quality of spectral mapping from speech to singing [1, 2, 16] are far from perfect that adversely affects the output singing quality. While the singing synthesis technique produces high quality and similarity singing vocals, it requires a large amount of training data for each target singer [8–11].

We note that it is always easier for a SVC system to learn from parallel training data [3,17–19]. However, the requirement of parallel singing training data limits the scope of the applications [3, 17–19]. There have been recent studies on the use of non-parallel singing data [20] where the target speaker identity needs to be learned through the conversion model. However, a model needs to be trained separately for each new target speaker. Others have studied the use of WaveNet for singing timbre conversion in an encoder-decoder manner, conditioned on the target speaker embedding look-up table [21].

The implementation of analysis-modification-synthesis pipeline usually modifies the parameters in frequency domain, therefore, require a vocoder, like WORLD [22] or STRAIGHT [23] to reconstruct the time-domain signals [3, 17–19]. However, these conventional vocoders are based on some prior assumptions, e.g. time invariant linear filter and source filter model, which cause quality degradation of synthesized voices [24]. Phase information of the generated samples is always discarded due to the simplification during mathematical formulation of the parametric model in the vocoders [24]. For example, the feature conversion approaches in SVC [3, 20] and STS conversion [16] suffer from the above problems while using a vocoder, which may cause quality degradation of generated singing voices.

To address these issues, trainable neural vocoders were proposed to synthesize time-domain audios such as WaveNet [25] and WaveRNN [26]. In particular, WaveNet was studied as a trainable vocoder that outperforms the traditional vocoder [22, 23, 27, 28] in voice conversion [27, 29]. However, there still exists a mismatch between the modified features from conversion model and the target features from natural vocies for neural vocoder synthesis. In this paper, we will study the use of trainable vocoder for singing waveform generation to reduce the feature mismatch problem.

We propose a novel SVG framework, that employs a WaveRNN [26] to map Phonetic PosteriorGram (PPG) [24, 30] and prosody features from singing voices, that are speaker/singer-independent, and the *speaker i-vector* from speech, directly to the singing waveform. Inspired by the recent studies on voice conversion and text-to-speech [30–36], we propose to use the *speaker i-vector* is to represent the speaker's voice identity. The proposed framework does not explicitly map the frames from one to another, but rather project the *speaker i-vector* as the encoded voice identity through the singing conversion and generation process. Once it is trained, the system works for any unseen target speakers. At run-time inference, what we need is to obtain template singing and some speech samples from the target speakers to derive a *speaker i-vector*, and present to the system.

The proposed framework requires neither the exact spoken lyrics from the source speaker/singer, nor parallel training data. In addition, we get rid of the frame alignment process that is required in template-based STS conversion and SVC . The proposed framework is also different from the technique in [20] as we do not need to train a target-dependent model, and is different from [21] where the system only works for a close set of speakers.

10.21437/Odyssey.2020-36

The main contributions of this paper include:

- We formulate a personalized SVG framework by proposing an integrated solution for both conversion and vocoding processes;
- We successfully adopt an integrated framework using WaveRNN to replace a two-step conversion-vocoding system, by which we reduce mismatches between training and testing;
- We do not require parallel training data and the associated time-alignment process.
- The use of i-vector allows the system to perform conversion for unseen target speakers without the need to re-train the system.

2. Conversion-Vocoding Pipeline

In SVG, we would like to carry over the speaker/singer independent information, such as lyrical content, prosody features from template. At the same time, we would like to project the voice quality or timbre of the target speaker/singer into the generated voice.

The conversion-vocoding pipeline, which includes conversion model and WaveRNN vocoder, is adopted in singing voice conversion and speech-to-singing conversion. First, we discuss how a conversion pipeline works for singing voice generation.

2.1. Conversion-Vocoding Pipeline

The pipeline employs a BLSTM network for the conversion of parameters, and WaveRNN vocoder for waveform generation, that we call a SVG-PL system.

We first decompose the input into speaker independent information (PPG, prosodic features) and speaker dependent information (*speaker i-vector*). The SVG-PL system is designed to take the two information from the singing template and the target speaker's speech samples as the input, and to generate personalized singing vocal as the output. As the Phonetic PosteriorGrams (PPG) coefficients are derived from the an Automatic Speech Recognition (ASR) system, that is designed to be speaker independent, we use PPG coefficients to represent the lyrical content. The prosody features, that are derived from the traditional vocoder, reflect the speaker/singer independent singing prosody that follows the sheet music. Therefore, the same prosody features should be carried over from template to target.

2.1.1. SVG-PL Training

In Figure 1 (a) and (b), we illustrate the training process of BLSTM model and WaveRNN vocoder, respectively. Singing PPG, fundamental frequency (F0), aperiodicity (AP) and Mel Cepstral Coeficients (MCCs) are first extracted from multi-singer singing. F0 and AP are used to represent the prosodic features. Multi-singer speaker identity features, i-vectors, are also obtained from their speech data. We use the acoustic features Mel MCCs as the BLSTM output. The prosodic features and MCCs can be derived from a traditional vocoder, such as WORLD vocoder [22].

A BLSTM model is then trained to map the input features, which consists of singing PPG, F0, AP and i-vectors, to their corresponding singing MCCs. While the WaveRNN vocoder is trained by conditioning on the acoustic features (singing MCCs, F0 and AP) augmented with the *speaker i-vector* for singing voice generation.

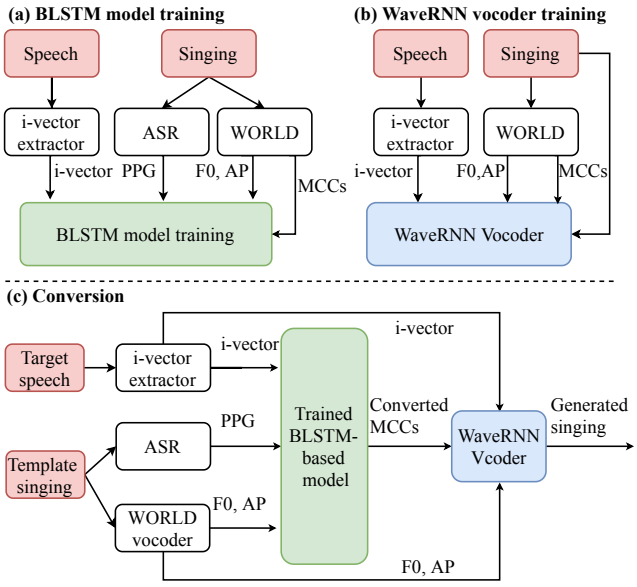

Figure 1: Block diagram of the training and run-time conversion of the BLSTM and WaveRNN vocoder pipeline (SVG-PL).

2.1.2. SVG-PL Conversion

At run-time, as shown in Figure 1 (c), target *speaker i-vector* is extracted from target speech samples, while PPG, F0 and AP, that are considered singer-independent, are obtained from template singing. Together they form the input to the trained BLSTM model. With the converted MCCs from BLSTM model, template prosodic features and target *speaker i-vector*, the WaveRNN vocoder generates the target singing.

2.2. WaveRNN vocoder

Next, we discuss WaveRNN and its vocoder implementation for SVG-PL. WaveRNN is a powerful recurrent network for speech synthesis [26]. The model mainly consists of the Gated Recurrent Unit (GRU) and the fully connected layers with softmax activation. In WaveRNN, it takes the previous audio sample s_{t-1} and conditions on $\mathbf{c}$, where $\mathbf{c}$ is the conditioning input, to model a discrete probability distribution P_{s_t} for sample generation, which is calculated by:

$$\begin{aligned} \mathbf{x}_t &= [s_{t-1}, \mathbf{c}] \\ \mathbf{u}_t &= \sigma(\mathbf{W}^{(u)}\mathbf{h}_{t-1} + \mathbf{U}^{(u)}\mathbf{x}_t) \\ \mathbf{r}_t &= \sigma\mathbf{W}^{(r)}\mathbf{h}_{t-1} + \mathbf{U}^{(r)}\mathbf{x}_t) \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{r}_t \circ (\mathbf{W}^{(h)}\mathbf{h}_{t-1}) + \mathbf{U}^{(h)}\mathbf{x}_t) \\ \mathbf{h}_t &= \mathbf{u}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \circ \tilde{\mathbf{h}}_t \\ P(s_t) &= \text{softmax}(\mathbf{W}_2\text{relu}(\mathbf{W}_1\mathbf{h}_t)) \end{aligned} \tag{1}$$

where $\mathbf{W}$, $\mathbf{U}$ represent the GRU weights, σ is the sigmoid activation function. The notation $\circ$ means an element-wise vector multiplication. $\mathbf{u}_t$, $\mathbf{r}_t$ and $\mathbf{h}_t$ are the gates in the GRU layer. The output sample s_t is obtained by sampling from the probability distribution $P(s_t)$. In the WaveRNN vocoder, the conditional input $\mathbf{c}$ can be acoustic features, prosodic features or speaker identity features. In the WaveRNN vocoder training phase of

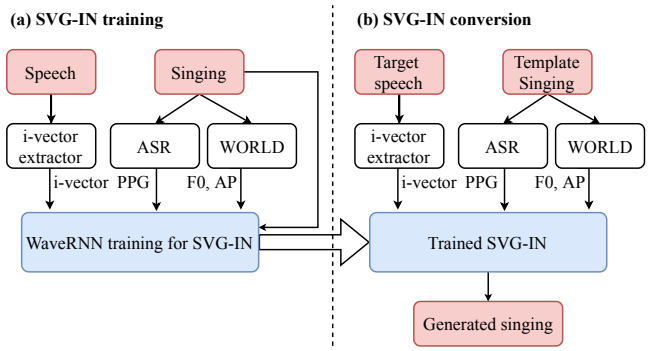

Figure 2: Block diagram of the training and run-time conversion for the proposed conversion-vocoding integrated framework (SVG-IN).

SVG-PL, we use the features as the conditional input $\mathbf{c}$. Therefore, $\mathbf{x_t}$ in Eq (1) can be re-written as:

$$\mathbf{x}_t = [s_{t-1}, \text{F0}_t, \text{AP}_t, \text{MCC}_t, \text{PPG}_t, \text{i-vector}] \quad (2)$$

3. Conversion-Vocoding Integrated Network

To improve the quality of generated singing, we propose an integrated solution to the conversion-vocoding pipeline, that we call the conversion-vocoding integrated network, or SVG-IN.

3.1. Motivation

While WaveRNN vocoder in SVG-PL is able to generate high quality target singing in general, it often generates unwanted noise at run-time inference. This is due to the mismatch between training and testing. During training, the input parameters to the vocoder are extracted from natural singing audios, while at run-time inference, the input parameters are generated by the BLSTM conversion model. Such mismatch has been addressed in the WaveNet vocoder [24, 37–39]. To overcome the mismatch during WaveRNN vocoder training and inference, we would like to investigate a novel WaveRNN implementation for mapping the singing PPG and speech i-vector directly to singing waveform.

3.2. Methodology

Similar to the SVG-PL framework, we use PPGs to represent the lyrical content, F0 and AP to represent the prosodic features. PPGs, F0 and AP are speaker independent. We also use *speaker i-vector* to represent the speaker individuality. Next, we present the proposed SVG-IN architecture and workflow.

3.2.1. SVG-IN Training

During training, as shown in Figure 2 (a), singing PPG and prosody features (F0 and AP) are first extracted from multi-singer singing data through ASR system and WORLD vocoder, respectively. Therefore, the singing template is represented by singing PPG, and two prosodic features, F0 and AP. The i-vectors are also extracted from the respective singers' speech data, so that the target speaker is represented by a *speaker i-vector* in the input. The local conditioning input of WaveRNN is then formulated by augmenting i-vectors and singing PPG, F0 and AP. Therefore, $\mathbf{x_t}$ in Eq (1) can be re-written as:

$$\mathbf{x}_t = [s_{t-1}, \text{F0}_t, \text{AP}_t, \text{PPG}_t, \text{i-vector}] \quad (3)$$

Compared with the training phase of SVG-PL, SVG-IN now presents one WaveRNN module to replace the conversion-vocoding pipeline. SVG-IN no longer requires the MCCs acoustic features as the intermediate representation.

3.2.2. SVG-IN Conversion

At run-time as shown in Figure 2 (b), given the target speech, we first extract target *speaker i-vector*. Singing PPG and prosody (F0 and AP) are also extracted from template singing through ASR and WORLD vocoder, respectively. Target singing is then generated by passing the augmented i-vector, singing PPG and prosody to the trained WaveRNN. In this way, the prosody and lyrical content of generated target singing is preserved from template singing. Different from the SVG-PL, SVG-IN doesn't use acoustic features MCCs as the intermediate representation, therefore, avoiding the possible mismatch arising from the intermediate features at run-time.

4. Experiments

4.1. Database

Two parallel speak-sing data databases, NUS-48E corpus [40] and NUS-HLT SLS corpus [41], were used for the experiments. The NUS-48E database contained 48 English songs from 12 singers/speakers, 6 males and 6 females, each singer having 4 songs. The NUS-HLT SLS database consists of 100 English songs from 10 singers, 5 females and 5 males, each singer having 10 songs.

We chose 18 speakers (9 male and 9 female), a total of 2,942 utterances for training. For testing, two unseen males and two unseen females were selected, with each male having 5 songs totalling 81 utterances while each female having 5 songs totalling 118 utterances. We used intra-gender speakers to serve as each other's template. All the objective and subjective results were reported based on the 20 songs (398 utterances) from the 4 target speakers. All speech and singing data were re-sampled at 16kHz.

4.2. Feature Extraction

We discuss the processes of extracting acoustic features, singing PPG and i-vectors for the experiments.

4.2.1. Acoustic feature extraction

The WORLD vocoder [22] was used to extract the spectrum (513-dim), AP (1-dim) and F0 (1-dim) for both speech and singing utterances with 5 ms frame shift. 40-dimensional MCCs were computed from the spectrum using Speech Signal Processing Toolkit (SPTK)[1].

4.2.2. Singing PPG extraction

The DNN-HMM ASR model was trained using the Kaldi toolkit [42]. The ASR model used for adaptive singing PPG extraction was trained with Wall Street Journal (WSJ) corpus [43]. The model consists of 5 hidden layers with 1,024 nodes in each layer, and a softmax output layer of 132 nodes. Hence, an adapted singing PPG vector is of 132 dimensions.

[1] https://sourceforge.net/projects/sp-tk/

4.2.3. i-vector extraction

An i-vector is a compact representation of a speaker model that contains the dominant speaker characteristics [44]. It is obtained using a factor analysis method that projects the high dimensional Gaussian mixture model mean super vector to a low dimensional vector. As this vector captures the dominant speaker information, it is referred to as identity vector or i-vector.

In this work, the background models for i-vector training were learned using Switchboard II corpus. We built 1024 component universal background model (UBM) and total variability matrix with 400 speaker factors to derive the 400-dimensional i-vectors [45]. Further, linear discriminant analysis was used for channel/session compensation and we reduced the dimension further to 60 for using it in all frameworks. We note that the utterances were downsampled to 8 kHz to match with the sampling rate of background data.

4.3. Experimental Setup

We now compare the SVG integrated network with the pipeline system, and one of its variants. As SVG-PL has a pipeline structure, that is similar to those for traditional template-based speech-to-singing conversion, we use SVG-PL as the reference baseline.

- **SVG-PL**: the BLSTM conversion model followed by WaveRNN vocoder, as described in Section 2. The input of BLSTM model consists of singing F0 (1-dim), singing AP (1-dim) with their delta and delta-delta coefficients, the voiced/unvoiced flag (1-dim), singing PPG (132-dim), singing energy (1-dim) and i-vectors (60-dim). The training input of WaveRNN vocoder includes singing F0 (1-dim), singing AP (1-dim) with their delta and delta-delta coefficients, singing MCCs (40-dim), singing energy (1-dim) and the voiced/unvoiced flag (1-dim).
- **SVG-WORLD**: a variant of SVG-PL, where the BLSTM conversion model is the same as that in SVG-PL, but the BLSTM model is followed by WORLD vocoder [22] instead of WaveRNN vocoder for singing generation. The setting of input parameters is the same as in SVG-PL. We note that the BLSTM network output dimension was 120, including singing MCCs (40-dim) with their delta and delta-delta coefficients. 513-dimensional spectrum were computed from converted singing MCCs using SPTK.
- **SVG-IN**: the proposed personalized singing generation using WaveRNN described in Section 3. The setting of input parameters is the same as in BLSTM model of SVG-PL.

The same i-vector extraction process of SVG-PL and SVG-IN were adopted in both training and run-time generation.

4.3.1. Model Training and Conversion

We used Merlin toolkit [46] for the training of the BLSTM model. The network consisted of two BLSTM layers with 256 nodes in each layer. The mini-batch size was set to 25. Adam optimizer was used for model training with the learning rate and momentum of 0.002 and 0.9, respectively. During the conversion phase in BLSTM model, the Maximum Likelihood Parameter Generation algorithm was then employed to refine the spectral parameter trajectory [47].

Both WaveRNN vocoder and the proposed SVG-IN model shared the same WaveRNN network architecture. With all singing signals sampled at 16kHz, the singing waveforms were quantized to 9-bit vectors by the μ-law in the WaveRNN module. To match the resolution of the singing waveform, the input features were first upsampled by the factor (2,4,10). The model architecture consisted of 2 GRU layers followed by 3 fully connected layers, all layers had 512 nodes. Batch size was set to 64, and the learning rate was $1 \times \mathrm{e}^{-4}$. Adam optimizer was used for both WaveRNN vocoder and SVG-IN training.

5. Evaluations

We conducted both objective and subjective evaluations in the experiments.

5.1. Objective Evaluations

We conducted the Root Mean Squared Error (RMSE) [24] to measure the distortion between the target and the converted singing. For each frame, RMSE is calculated as follows,

$$\text{RMSE[dB]} = \sqrt{\frac{1}{F}\sum_{f=1}^{F}(20\log_{10}(\frac{|Y(f)|}{|Y_g(f)|}))^2} \qquad (4)$$

where F is the number of total frequency bins, and $|Y(f)|$ and $|Y_g(f)|$ are magnitude features of the target and the generated singing at the fth frequency bin, respectively. The lower RMSE value indicates smaller distortion. Table 1 shows the average RMSE results over all the evaluation utterances for the three systems. We first assessed the effect of intermediate features to WaveRNN generation. We observe that SVG-IN outperforms SVG-PL in terms of RMSE (15.75 dB vs 17.75 dB). This suggests that SVG-IN has reduced the mismatch arising from the intermediate features in SVG-PL. While comparing SVG-SI with SVG-WORLD, it is showed that SVG-WORLD (15.00 dB) gives slightly lower RMSE value than SVG-IN (15.75 dB). We note that RMSE measures how close the generated singing is to the target singing.

Table 1: A comparison of RMSE (dB) results among SVG-PL, SVG-WORLD and SVG-IN.

System	RMSE
SVG-PL	17.75
SVG-WORLD	15.00
SVG-IN	15.75

As SVG-IN operates in time domain, its RMSE in frequency domain is an indirect measurement, that may lead to a higher RMSE than SVG-WORLD. Similar results were also reported in [24,27,48] where conventional vocoders typically give a better objective measure than that of trainable neural vocoders.

5.2. Subjective Evaluations

AB preference and XAB preference tests were also conducted for subjective evaluations. 12 listeners comprising a mix of native and non-native speakers with normal hearing abilities aging from 20 to 35 participated in all the tests. In AB preference

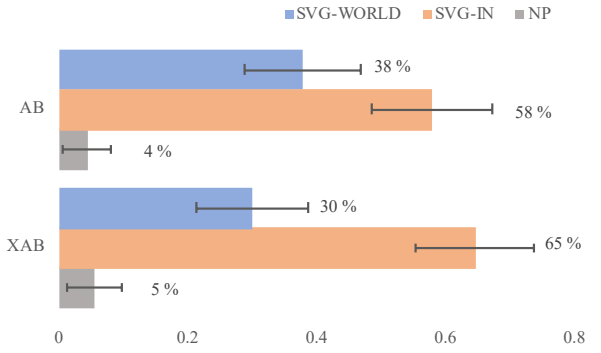

Figure 3: AB and XAB preference results with 95% confidence intervals for measuring quality and similarity of SVG-IN and SVG-WORLD. NP stands for no preference.

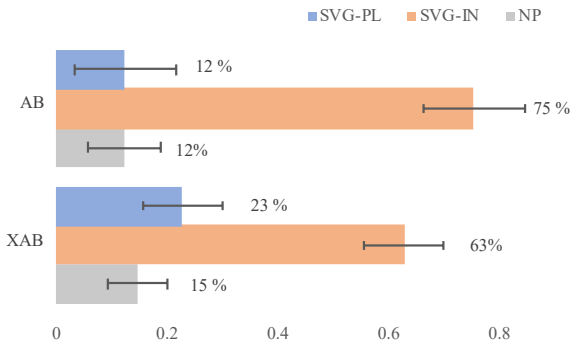

Figure 4: AB and XAB preference results with 95% confidence intervals for measuring quality and similarity of SVG-IN and SVG-PL. NP stands for no preference.

tests, one singing sample was picked from system A and another was from system B. Listeners were asked to select the better one in terms of the quality and naturalness of the generated singing. The XAB preference tests were employed to evaluate the speaker similarity between generated singing and target singing, where listeners were asked to to choose if the sample from system A or system B is more similar to the target singing X. We randomly selected 20 out of 398 generated samples for system A and another corresponding 20 out of 398 generated samples from system B to form a pair. Each listener listened to 20 unique pairs of samples.

5.2.1. SVG-IN vs SVG-WORLD

We first compare SVG-IN with SVG-WORLD as showed in Figure 3. The results of AB preference tests show that the proposed SVG-IN outperforms SVG-WORLD with the preference of 58% and 38%, respectively. This indicates that the quality and naturalness of generated singing is enhanced due to the usage of WaveRNN. Moreover, the results of XAB preference tests are consistent with that of AB preference tests, where the proposed SVG-IN (65%) achieves better similarity performance than the SVG-WORLD (30%). Benefiting from the direct waveform generation process, the integrated WaveRNN system is able to preserve user's speaker identity better than the system with traditional parametric vocoders. Overall, the subjective evaluations confirm the effectiveness of the integrated network for SVG.

5.2.2. SVG-IN vs SVG-PL

We further report the results of AB and XAB preference tests for SVG-IN and SVG-PL in Figure 4. It is found that SVG-IN significantly outperforms SVG-PL both in terms of quality and naturalness (with SVG-IN 75% .vs SVG-PL 12%). This clearly demonstrates the superiority of the SVG-IN in overcoming the feature mismatches between WaveRNN training and conversion processes, thereby providing higher quality singing vocals. It is also observed that the SVG-IN delivers a relative improvement of 40% in XAB preference test over SVG-PL, that confirms its efficacy in capturing and distinguishing speakers in the integrated network. The subjective evaluations confirm the effectiveness of direct feature-to-waveform generation that avoids the mismatch problem in the two-step training in the conversion-vocoding pipeline.

5.3. Summary of Evaluation Results

The objective and subjective evaluations confirm that SVG-IN is superior to the SVG-PL pipeline and SVG-WORLD baseline in terms of quality, naturalness and similarity. In addition, the proposed SVG-IN allows for singing voice generation for any unseen speakers, as long as we obtain sample speech from the target speakers. SVG-IN provides great flexibility for personalized singing voice generation applications. The generated singing samples from different systems are available in the following website[2].

6. Conclusion

This paper presents a novel personalized singing voice generation approach using WaveRNN. The SVG framework does not require parallel singing data or any target singing during training, therefore, no frame alignment is needed. The integrated network solution (SVG-IN) avoids features mismatch issue between WaveRNN training and run-time conversion, which leads to higher quality and similarity singing outputs. In this study, we extract *speaker i-vector* from speech samples of target speaker to represent the voice identity of a person. We expect the system to work as well if it is presented with singing samples from target singer. The experiment results conducted on NUS-48E and NUS-HLT SLS corpora confirm the effectiveness of the proposed SVG-IN in terms of quality and similarity.

7. Acknowledgment

This research work is supported by Academic Research Council, Ministry of Education (ARC, MOE). Grant: MOE2018-T2-2-127. Title: Learning Generative and Parameterized Interactive Sequence Models with RNNs. It is also supported by Human-Robot Interaction Phase 1 (Grant No. 192 25 00054), National Research Foundation Singapore under the National Robotics Programme, and (Award Number: AISG-100E-2018-006) under the AI Singapore Programme, and Programmatic Grant No. A18A2b0046 (Human Robot Collaborative AI for AME) from the Singapore Government's Research, Innovation and Enterprise 2020 plan in the Advanced Manufacturing and Engineering domain.

[2]http://xiaoxue1117.github.io/odysseysample

8. References

[1] Ling Cen, Minghui Dong, and Paul Chan, "Template-based personalized singing voice synthesis," in *IEEE ICASSP*, 2012, pp. 4509–4512.

[2] Karthika Vijayan, Minghui Dong, and Haizhou Li, "A dual alignment scheme for improved speech-to-singing voice conversion," in *IEEE APSIPA ASC*, 2017, pp. 1547–1555.

[3] Berrak Sisman, Karthika Vijayan, Minghui Dong, and Haizhou Li, "SINGAN: Singing voice conversion with generative adversarial networks," *in IEEE APSIPA ASC*, pp. 112–118, 2019.

[4] Xavier Rodet, "Synthesis and processing of the singing voice," in *in Proc. 1st IEEE MPCA*. Citeseer, 2002, pp. 15–31.

[5] Hideki Kenmochi and Hayato Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation," in *Interspeech*, 2007, pp. 4009–4010.

[6] Keijiro Saino, Makoto Tachibana, and Hideki Kenmochi, "A singing style modeling system for singing voice synthesizers," in *Interspeech*, 2010, pp. 2894–2897.

[7] Tomoyasu Nakano and Masataka Goto, "Vocalistener2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics," in *IEEE ICASSP*, 2011, pp. 453–456.

[8] Masanari Nishimura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Singing voice synthesis based on deep neural networks.," in *Interspeech*, 2016, pp. 2478–2482.

[9] Juntae Kim, Heejin Choi, Jinuk Park, Sangjin Kim, Jongjin Kim, and Minsoo Hahn, "Korean singing voice synthesis system based on an LSTM recurrent neural network," in *Interspeech*, 2018, pp. 1551–1555.

[10] Merlijn Blaauw and Jordi Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, pp. 1313, 2017.

[11] Juheon Lee, Hyeong-Seok Choi, Chang-Bin Jeon, Junghyun Koo, and Kyogu Lee, "Adversarially trained end-to-end Korean singing voice synthesis system," *in Interspeech*, pp. 2588–2592, 2019.

[12] Karthika Vijayan, Xiaoxue Gao, and Haizhou Li, "Analysis of speech and singing signals for temporal alignment," in *IEEE APSIPA ASC*, 2018, pp. 1893–1898.

[13] Bidisha Sharma and Haizhou Li, "A combination of model-based and feature-based strategy for speech-to-singing alignment," *in Interspeech*, pp. 624–628, 2019.

[14] Siu Wa Lee and Minghui Dong, "Singing voice synthesis: Singer-dependent vibrato modeling and coherent processing of spectral envelope," in *Interspeech*, 2011, pp. 2001–2004.

[15] Siu Wa Lee, Zhizheng Wu, Minghui Dong, Xiaohai Tian, and Haizhou Li, "A comparative study of spectral transformation techniques for singing voice synthesis," in *Interspeech*, 2014, pp. 2499–2503.

[16] Xiaoxue Gao, Xiaohai Tian, Rohan Kumar Das, Yi Zhou, and Haizhou Li, "Speaker-independent spectral mapping for speech-to-singing conversion," *in IEEE APSIPA ASC*, pp. 159–164, 2019.

[17] Hironori Doi, Tomoki Toda, Tomoyasu Nakano, Masataka Goto, and Satoshi Nakamura, "Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system," in *IEEE APSIPA ASC*, 2012, pp. 1–6.

[18] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura, "Statistical singing voice conversion based on direct waveform modification with global variance," in *Interspeech*, 2015, pp. 2754–2758.

[19] Kazuhiro Kobayashi, Tomoki Toda, and Satoshi Nakamura, "Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential," *Speech Communication*, vol. 99, pp. 211–220, 2018.

[20] Xin Chen, Wei Chu, Jinxi Guo, and Ning Xu, "Singing voice conversion with non-parallel data," in *IEEE MIPR*, 2019, pp. 292–296.

[21] Eliya Nachmani and Lior Wolf, "Unsupervised singing voice conversion," *in Interspeech*, pp. 2583–2587, 2019.

[22] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[23] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De 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.

[24] Xiaohai Tian, Eng Siong Chng, and Haizhou Li, "A Speaker-Dependent WaveNet for Voice Conversion with Non-Parallel Data," in *Interspeech*, 2019, pp. 201–205.

[25] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[26] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[27] Tomoki Hayashi, Akira Tamamori, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "An investigation of multi-speaker training for WaveNet vocoder," in *IEEE ASRU Workshop*, 2017, pp. 712–718.

[28] Nagaraj Adiga, Vassilis Tsiaras, and Yannis Stylianou, "On the use of WaveNet as a statistical vocoder," in *IEEE ICASSP*, 2018, pp. 5674–5678.

[29] Li-Juan Liu, Zhen-Hua Ling, Yuan Jiang, Ming Zhou, and Li-Rong Dai, "WaveNet vocoder with limited training data for voice conversion.," in *Interspeech*, 2018, pp. 1983–1987.

[30] Yi Zhou, Xiaohai Tian, Haihua Xu, Rohan Kumar Das, and Haizhou Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *IEEE ICASSP*, 2019, pp. 6790–6794.

[31] Junichi Yamagishi, Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, and Takao Kobayashi, "A training method of average voice model for HMM-based speech synthesis," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 86, no. 8, pp. 1956–1963, 2003.

[32] Junichi Yamagishi, Takashi Nose, Heiga Zen, Zhen-Hua Ling, Tomoki Toda, Keiichi Tokuda, Simon King, and Steve Renals, "Robust speaker-adaptive HMM-based text-to-speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1208–1230, 2009.

[33] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He, "Multi-speaker modeling and speaker adaptation for DNN-based tts synthesis," in *IEEE ICASSP*, 2015, pp. 4475–4479.

[34] Jie Wu, Zhizheng Wu, and Lei Xie, "On the use of i-vectors and average voice model for voice conversion without parallel data," in *IEEE APSIPA ASC*, 2016, pp. 1–6.

[35] Xiaohai Tian, Junchao Wang, Haihua Xu, Eng Siong Chng, and Haizhou Li, "Average modeling approach to voice conversion with non-parallel data," in *Odyssey The Speaker and Language Recognition Workshop*, 2018, pp. 227–232.

[36] Tetsuya Hashimoto, Daisuke Saito, and Nobuaki Minematsu, "Many-to-many and completely parallel-data-free voice conversion based on eigenspace DNN," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 332–341, 2018.

[37] Yi-Chiao Wu, Patrick Lumban Tobing, Tomoki Hayashi, Kazuhiro Kobayashi, and Tomoki Toda, "The NU non-parallel voice conversion system for the voice conversion challenge 2018," *WORLD*, vol. 2, no. m3, pp. a1, 2018.

[38] Yi-Chiao Wu, Kazuhiro Kobayashi, Tomoki Hayashi, Patrick Lumban Tobing, and Tomoki Toda, "Collapsed speech segment detection and suppression for WaveNet vocoder," *arXiv preprint arXiv:1804.11055*, 2018.

[39] Kazuhiro Kobayashi, Tomoki Hayashi, Akira Tamamori, and Tomoki Toda, "Statistical voice conversion with WaveNet-based waveform generation.," in *Interspeech*, 2017, pp. 1138–1142.

[40] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang, "The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *IEEE APSIPA ASC*, 2013, pp. 1–9.

[41] Xiaoxue Gao, Berrak Sisman, Rohan Kumar Das, and Karthika Vijayan, "NUS-HLT spoken lyrics and singing (SLS) corpus," in *IEEE ICOT*, 2018, pp. 1–6.

[42] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE ASRU Workshop*, 2011, number EPFL-CONF-192584.

[43] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based CSR corpus," in *Proc. the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[44] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[45] Rohan Kumar Das and S. R. M. Prasanna, "Exploring different attributes of source information for speaker verification with limited test data," *The Journal of the Acoustical Society of America*, vol. 140, no. 1, pp. 184–190, 2016.

[46] Zhizheng Wu, Oliver Watts, and Simon King, "Merlin: An open source neural network speech synthesis system.," in *the 9th ISCA SSW*, 2016, pp. 202–207.

[47] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *IEEE ICASSP*, 2000, pp. 1315–1318.

[48] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "Speaker-Dependent WaveNet vocoder.," in *Interspeech*, 2017, pp. 1118–1122.