A COMPARISON OF RECENT WAVEFORM GENERATION AND ACOUSTIC MODELING METHODS FOR NEURAL-NETWORK-BASED SPEECH SYNTHESIS

Xin Wang[1], *Jaime Lorenzo-Trueba*[1], *Shinji Takaki*[1], *Lauri Juvela*[2], *Junichi Yamagishi*[1*]

[1]National Institute of Informatics, Japan
[2]Aalto University, Finland

wangxin@nii.ac.jp, jaime@nii.ac.jp, takaki@nii.ac.jp, lauri.juvela@aalto.fi, jyamagis@nii.ac.jp

ABSTRACT

Recent advances in speech synthesis suggest that limitations such as the lossy nature of the amplitude spectrum with minimum phase approximation and the over-smoothing effect in acoustic modeling can be overcome by using advanced machine learning approaches. In this paper, we build a framework in which we can fairly compare new vocoding and acoustic modeling techniques with conventional approaches by means of a large scale crowdsourced evaluation. Results on acoustic models showed that generative adversarial networks and an autoregressive (AR) model performed better than a normal recurrent network and the AR model performed best. Evaluation on vocoders by using the same AR acoustic model demonstrated that a Wavenet vocoder outperformed classical source-filter-based vocoders. Particularly, generated speech waveforms from the combination of AR acoustic model and Wavenet vocoder achieved a similar score of speech quality to vocoded speech.

Index Terms— speech synthesis, deep learning, Wavenet, general adversarial network, autoregressive neural network

1. INTRODUCTION

Text-to-speech (TTS) synthesis aims at converting a text string into a speech waveform. A conventional TTS pipeline normally includes a frond-end text-analyzer and a back-end speech synthesizer, each of which may include multiple modules for specific tasks. For example, the back-end based on statistical parametric speech synthesis (SPSS) uses statistical acoustic models to map linguistic features from the text-analyzer into compact acoustic features. It then uses a vocoder to generate a waveform from the acoustic features.

Recent TTS systems have well adopted to the impact of deep learning. First, frameworks different from the established TTS pipeline are defined. One framework merges the text-analyzer and acoustic models into a single neural network (NN), where the input text is directly mapped into acoustic features, e.g., by Tactron [1] and Char2wav [2]. Another framework is the unified back-end represented by Wavenet [3] that directly converts the linguistic features into a waveform. There are also novel NN back-ends without the use of normal acoustic features, vocoders [4, 5], or duration model [6]. In parallel to these new frameworks, modules for the conventional TTS pipeline have also been developed. For instance, acoustic models based on the generative adversarial network (GAN) [7] and autoregressive (AR) NN [8] have been reported to alleviate the over-smoothing effect in acoustic modeling.

*This work was partially supported by MEXT KAKENHI Grant Numbers (15H01686, 16H06302, 17H04687), and JST ACT-I Grant Number (JPMJPR16UG).

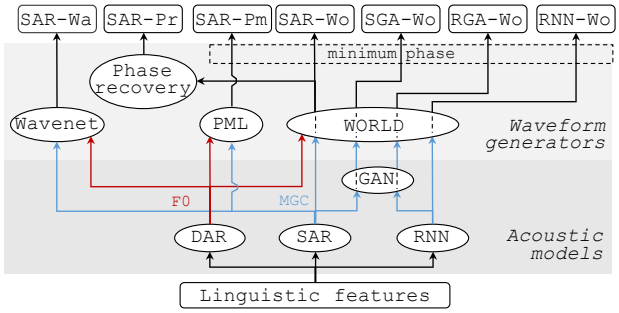

Fig. 1: Flowchart for seven speech synthesis methods compared in this study. This figure hides the band aperiodicity (BAP) features and RNN that generates noise mask for PML vocoder.

There is also the Wavenet-based vocoder [9] that uses deep learning rather than signal processing methods to attain waveform generation.

Given the recent progress with TTS, we think it is important to understand the pros and cons of the new methods on the basis of common conditions. As an initial step, we compared the new methods that could easily be plugged into the SPSS-based back-end of the classical TTS pipeline. These methods were divided into two groups, and thus the comparison included two parts. The first part compared acoustic models based on four types of NN, including the normal recurrent neural network (RNN), shallow AR-based RNN (SAR), and two GAN-based postfilter networks [7] that were attached to the RNN and SAR. The second part compared waveforms generation techniques, including the high-quality source-filter vocoder WORLD [10], another synthesizer based on the log-domain pulse model (PML) [11], WORLD plus the Griffin-Lim algorithm for phase recovery [12], and the Wavenet-based vocoder.

Section 2 explains the details on the selected methods used for comparison. Section 3 explains the experiments and Section 4 draws conclusions.

2. SPEECH SYNTHESIS METHODS

This comparison study was based on perceptual evaluation of synthetic speech. The selected acoustic models and waveform generation methods were combined into the seven speech synthesis methods given in Fig. 1, where `SAR-Wa`, `SAR-Pm`, `SAR-Pr`, and `SAR-Wo` use the same acoustic models but different waveform generation modules, while `SAR-Wo`, `GAS-Wo`, `GAR-Wo`, and `RNN-Wo` use the same WORLD vocoder but differ in the acoustic model to generate the Mel-generalized coefficients (MGC) and band aperiodicity (BAP). The F0 for all the methods is generated by a deep AR (DAR) model [13]. The oracle duration is used for all the methods. Neither formant enhancement nor MLPG [14] was used.

2.1. Waveform generation

2.1.1. Relationship between acoustic features and waveforms

A vocoder in a conventional speech synthesis method synthesizes the waveform given the generated acoustic features by the acoustic model. However, the synthesized waveform may be unnatural even if the acoustic model is perfect because the acoustic features may be imperfectly defined. For example, to extract cepstral features from the waveform, the first step is to apply discrete Fourier transform (DFT) to the windowed waveform and then acquire the complex spectrum, $\boldsymbol{s}_n = [s_{n,1}, \cdots, s_{n,F}] \in \mathbb{C}^F$. Here, $s_{n,f} = a_{n,f} + ib_{n,f} \in \mathbb{C}$ is a complex number, n is the index of a speech frame, and $f \in [1, F]$ is the index of a frequency bin. The next step is to compute the power spectrum, $\boldsymbol{p}_n = [p_{n,1}, \cdots, p_{n,F}] = diag(\boldsymbol{s}_n^H \boldsymbol{s}_n)$, where $\boldsymbol{p}_n \in \mathbb{R}^F$. Note that H is the Hermitian transpose, and then $p_{n,f} = |s_{n,f}|^2 = {a_{n,f}}^2 + {b_{n,f}}^2 \in \mathbb{R}$. Finally, cepstral feature $\boldsymbol{c}_n \in \mathbb{R}^M$, where $M < F$, is acquired after inverse DFT plus filtering and frequency warping on log-amplitude spectrum $[\log\sqrt{p_{n,1}}, \cdots, \log\sqrt{p_{n,F}}]$. Notice that $\boldsymbol{p}_n$ encodes the amplitude of $\boldsymbol{s}_n$ while ignoring the phase. Therefore, $\boldsymbol{s}_n$ and the waveform cannot be accurately reproduced only given $\boldsymbol{c}_n$.

Conventional vocoders assume that $\boldsymbol{s}_n$ has a minimum phase in order to approximately recover $\boldsymbol{s}_n$ from $\boldsymbol{c}_n$. However, the speech waveform is not merely a minimum-phase signal. One solution is to model the phase as additional acoustic features [15, 16]. Another way is to directly model complex-valued $\boldsymbol{s}_n$ by using complex-valued NN [17] or restricted Boltzman machine [18]. Alternatively, a statistical model such as the Wavenet-based vocoder [9] can be used rather than using a deterministic vocoder. Such a vocoder may learn the phase information not encoded by the acoustic features.

This work used a conventional vocoder called WORLD [10] as the baseline. It then included a phase-recovery technique [12], a waveform synthesizer based on a log-domain pulse model [11], and a Wavenet-based vocoder for comparison. Complex-valued approaches may be included in future work.

2.1.2. Deterministic vocoders

The WORLD vocoder [10] is similar to the legacy STRAIGHT vocoder [19] and uses a source-filter-based speech production model with mixed excitation. It also assumes a minimum phase for the spectrum. By using the procedure in Sec. 2.1.1 reversely, WORLD converts the cepstral features, $\boldsymbol{c}_n$, given by the acoustic model into a linear amplitude spectrum. Then, WORLD creates the excitation signal by mixing a pulse and a noise signal in the frequency domain, where each frequency band is weighted by a BAP value. Finally, it generates a speech waveform based on the source-filter model.

Since WORLD assumes a minimum phase, this work included a phase-recovery method that may enhance the phase of the generated waveform by WORLD. This method re-computes the amplitude spectrum given the generated waveform and then applies the Griffin-Lim phase-recovery algorithm. Note that this method is different from that in our previous work [5, 20] where phase-recovery was directly conducted on the generated amplitude spectrum.

This work also included the pulse model in the log-domain (PML) [11] as another waveform generator. One advantage of PML is that it avoids voicing decision per frame and may alleviate the perceptual degradation caused by voicing decision errors. In addition, PML may better represent noise in voiced segments. The PML and WORLD in this work used the same generated spectrum as input. The noise mask used by PML is generated by another RNN given input linguistic features, which it is not indicated in Fig. 1.

2.1.3. Data-driven vocoder

Besides the deterministic vocoders, this work included a statistical vocoder based on Wavenet [9, 21]. Suppose $\boldsymbol{o}_t$ is a one-hot vector representing the quantized waveform sampling point at time t, the Wavenet vocoder infers the distribution $P(\boldsymbol{o}_t|\boldsymbol{o}_{t-R:t-1}, \boldsymbol{a}_t)$ given the acoustic feature $\boldsymbol{a}_t$ and previous observations $\boldsymbol{o}_{t-R:t-1} = \{\boldsymbol{o}_{t-R}, \cdots, \boldsymbol{o}_{t-1}\}$, where R is the receptive field of the network. In this work, $\boldsymbol{a}_t$ contained the F0 and MGC. Note that, while natural $\boldsymbol{a}_t$ is used to train the vocoder, $\widehat{\boldsymbol{a}}_t$ generated by the acoustic model is used for waveform generation.

During generation, $\widehat{\boldsymbol{o}}_t$ is usually obtained by random sampling, i.e., $\widehat{\boldsymbol{o}}_t \sim P(\boldsymbol{o}_t|\widehat{\boldsymbol{o}}_{t-R:t-1}, \widehat{\boldsymbol{a}}_t)$. However, we found that the synthetic waveform sounded better when samples in the voiced segments were generated as $\widehat{\boldsymbol{o}}_t = \arg\max_{\boldsymbol{o}_t} p(\boldsymbol{o}_t|\widehat{\boldsymbol{o}}_{t-R:t-1}, \widehat{\boldsymbol{a}}_t)$. This method is referred to as the *greedy generation* method. Fig. 4 plots the spectrogram and the instantaneous frequency (IF) [22, 23] of generated waveforms from the two methods, i.e., `SAR-Wa` (greedy) and `SAR-Wa` (random). Compared with random sampling, the greedy method generated more regular IF patterns in voiced segments. It perceptually made the waveforms sound less trembling. One reason for this may be that the 'best' generated $\widehat{\boldsymbol{o}}_t$ was temporally more consistent with $\widehat{\boldsymbol{o}}_{t-R:t-1}$. Note that samples in the unvoiced parts were still randomly drawn, and the voiced/unvoiced state for each time could be inferred from the F0 in $\widehat{\boldsymbol{a}}_t$.

2.2. Neural network acoustic models

The NN-based acoustic models included in this work aims at inferring the distribution of the acoustic feature sequence $\boldsymbol{a}_{1:N} = \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_N\}$ conditioned on the linguistic feature $\boldsymbol{l}_{1:N} = \{\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N\}$ in N frames. The baseline RNN, whether it has a recurrent output layer or not, defines the distribution as

$$p(\boldsymbol{a}_{1:N}|\boldsymbol{l}_{1:N}) = \prod_{n=1}^{N} p(\boldsymbol{a}_n|\boldsymbol{l}_{1:N}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{a}_n; \boldsymbol{h}_n, \boldsymbol{I}), \quad (1)$$

where $\mathcal{N}(\cdot)$ is the Gaussian distribution, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{h}_n = \mathcal{H}_{\boldsymbol{\Theta}}(\boldsymbol{l}_{1:N}, n)$ is the outcome of the output layer at frame n, and $\boldsymbol{\Theta}$ is the network's weight. For generation, $\widehat{\boldsymbol{a}}_{1:N}$ can be acquired by using a mean-based method that sets $\widehat{\boldsymbol{a}}_{1:N} = \boldsymbol{h}_{1:N}$.

A shortcoming of RNN is that the across-frame dependency in $\boldsymbol{a}_{1:N}$ is ignored. Hence, this work included a model that takes into account the dependency in a causal direction [8]. This model, which is called shallow AR (SAR), defines the distribution as:

$$p(\boldsymbol{a}_{1:N}|\boldsymbol{l}_{1:N}) = \prod_{n=1}^{N} p(\boldsymbol{a}_n|\boldsymbol{a}_{n-K:n-1}, \boldsymbol{l}_{1:N}) \quad (2)$$

$$= \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{a}_n; \boldsymbol{h}_n + \mathcal{F}_{\Phi}(\boldsymbol{a}_{n-K:n-1}), \boldsymbol{I}). \quad (3)$$

This model uses $\mathcal{F}_{\boldsymbol{\Phi}}(\boldsymbol{a}_{n-K:n-1}) = \sum_{k=1}^{K} \boldsymbol{\beta}_k \odot \boldsymbol{a}_{n-k} + \boldsymbol{\gamma}$ to merge the acoustic features in the previous K frames and then changes the distribution for frame n, which builds the causal across-frame dependency. Another deep AR (DAR) model similar to SAR can be defined [13]. DAR in this work was only used to model the quantized F0 for all the experimental methods, and thus is not explained here.

As RNN and SAR are trained by using the maximum-likelihood criterion and then generate by using the mean-based method, they may not produce acoustic features with natural textures. Therefore, this work also included the GAN-based postfilter [7] to enhance RNN and SAR. The GAN discriminators in this work did not crop the input acoustic features into small patches, which is different from original work. Instead, they made a true-false judgment every frame.

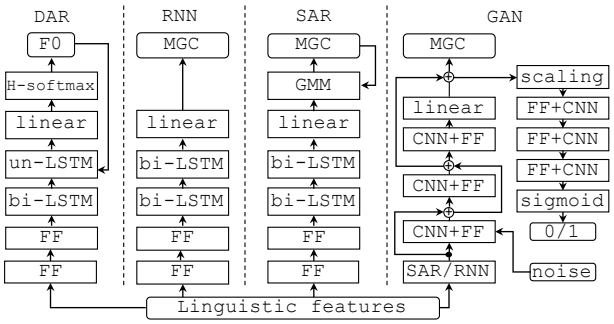

Fig. 2: Structure of acoustic models. Linear, FF, H-softmax, and un- and bi-LSTM correspond to linear transformation, feedforward layer using $tanh$, hierarchical softmax [13], and uni- and bi-directional LSTM layers. GMM is Gaussian mixture model. Scaling layer in GAN scales each dimension of input features.

3. EXPERIMENTS

3.1. Corpus and features

This work used a Japanese speech corpus [24] of neutral reading speech uttered by a female speaker. The duration is 50 hours, and the number of utterance is 30,016, out of which 500 were randomly selected as a validation set and another 500 were used as a test set. Linguistic features were extracted by using OpenJTalk [25]. The dimension of these features vectors was 389. Acoustic features were extracted at a frame rate of 200 Hz (5 ms) from the 48 kHz waveforms. MGC and BAP were extracted by using WORLD. The dimensions for MGC were 60 and those for BAP were 25. The F0 was extracted by an assembly of multiple pitch trackers and then quantized into 255 levels for quantized F0 modeling [13].

3.2. Neural network configuration

The network structures of the acoustic models are plotted in Fig. 2. In DAR, RNN and SAR, the size of feedforward (FF), bi- and uni-directional LSTM layer was 512, 256, and 128. The size of the linear layer depended on the output features' dimensions. For SAR, the output layer included a Gaussian distribution for MGC with the AR parameter $K = 1$ and another Gaussian for BAP with $K = 0$. In GAN, all the CNN layers had 256 output channels and conducted 1D convolution. Each FF layer in the GAN generator changed the dimension of a CNN's output before skip-add operation. To reduce instability in low-dimensional MGCs, the input MGC to the discriminator was scaled element-wisely. The scaling weight was 0.001 for the first five MGC dimensions, 0.01 for the next five dimensions, and 1 for the rest. The weight was 0 for BAP.

The Wavenet vocoder works at a sampling rate of 16kHz. The μ-law companded waveform is quantized into 10 bits per sample. Similar to the literature [9], the network consists of a linear projection input layer, 40 blocks for dilated convolution, and a post-processing block. The k-th dilation block has a dilation size of $2^{\mathrm{mod}(k,10)}$, where $\mathrm{mod}(\cdot)$ is modulo operation. The acoustic features, which are fed to every dilated convolution block, contains MGC and quantized F0. Natural acoustic features are used for training while generated MGC and F0 are used during generation.

All the acoustic models and Wavenet are implemented on a modified CURRENNT toolkit [26]. This toolkit and training recipes can be found online (`http://tonywangx.github.io`).

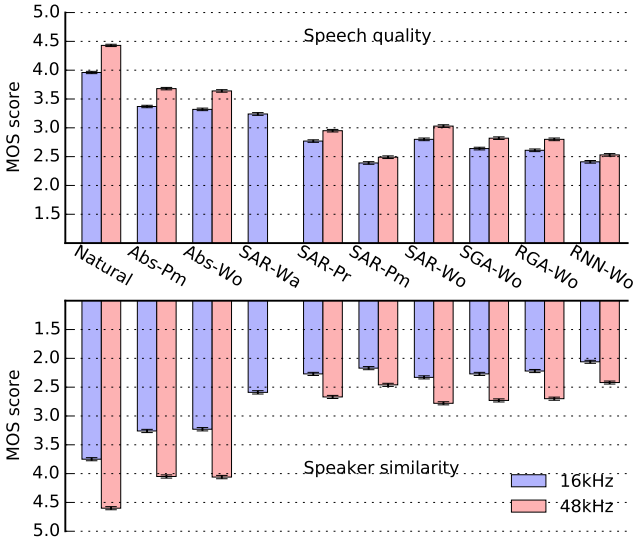

Fig. 3: Results in MOS scale. `AbS-Pm` and `AbS-Wo` correspond to vocoded speech by using PML and WORLD vocoders. Error bars represent two-tailed Student 95% confidence intervals.

3.3. Evaluation environment

The seven speech synthesis methods in Fig.1 were evaluated in terms of speech quality and speaker similarity. Natural and vocoded speech from WORLD and PML were also included in the evaluation. All the systems except for the Wavenet-based `SAR-Wa` were rated at sampling rates of 48 and 16 kHz. The speech samples of these systems were natural or generated at 48 kHz, and the 16 kHz samples were down-sampled from these 48 kHz samples. `SAR-WA` only generated samples of 16 kHz. All samples were normalized to -26 dBov, and a total of 19 groups of speech samples were evaluated.

The evaluation was crowdsourced online. Each evaluation set had 19 screens, i.e., one for each system. The order of systems in each set was randomly shuffled. The evaluators must answer two questions on each screen. First, they listened to a sample of the system under evaluation and rated the naturalness on a 1-to-5 MOS scale. They then rated the similarity of that test sample to the natural 48 kHz sample on a 1-to-5 MOS scale. The participants were allowed to replay the samples. The samples in one set were synthesized for the same text randomly selected from the test set.

3.4. Results and discussion

We collected a total of 1500 evaluation sets, i.e., 1500 scores for each system. A total of 235 native Japanese listeners participated, with an average of 6.4 sets per person. The statistical analysis was based on unpaired t-tests with a 95% confidence margin and Holm-Bonferroni compensation. The results are plotted in Fig. 3.

On acoustic modeling, the comparisons of the speech quality score among `RNN-Wo`, `RGA-Wo`, `SAR-Wo`, and `SGA-Wo` indicated that SAR (3.03) $>$ SGA (2.82) $\sim$ RGA (2.81) $>$ RNN (2.53) for both 48 kHz and 16 kHz. The same ordering can be seen in terms of speaker similarity. RNN's unsatisfactory performance may be due to the over-smoothing effect on the generated MGC, which muffled the synthetic speech. This effect is indicated by the global variance (GV) plotted in Fig. 5, where the GV of the generated MGC from RNN was smaller than that of the other models. The performance of SAR is consistent with that in our previous work [8], which indicated

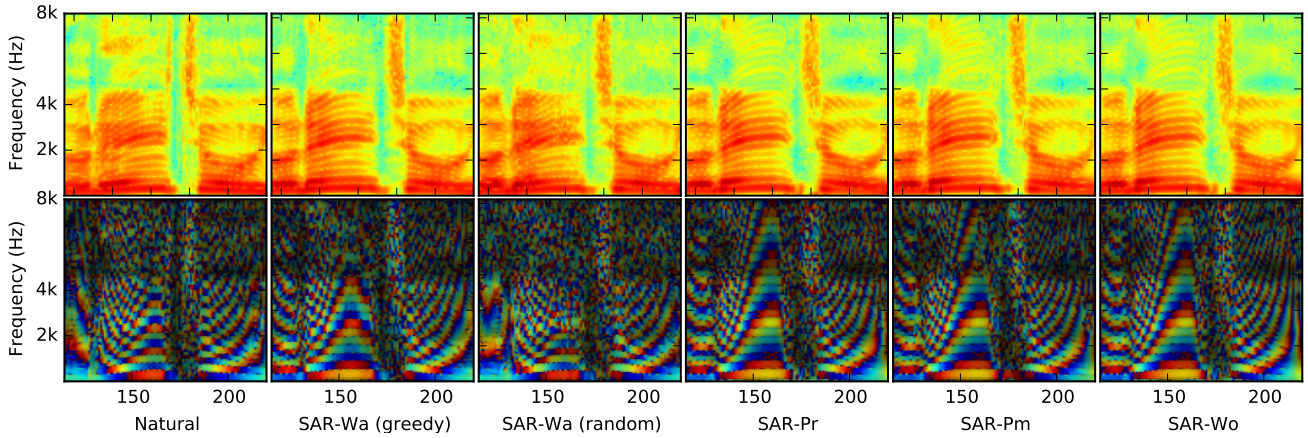

Fig. 4: Spectrogram (top) and instantaneous frequency (bottom) of natural speech sample, synthetic samples from SAR-Wa using greedy generation, SAR-Wa using random sampling, SAR-Pr, SAR-Pm and SAR-Wo. Test utterance ID is AOZORAR_03372_T01.

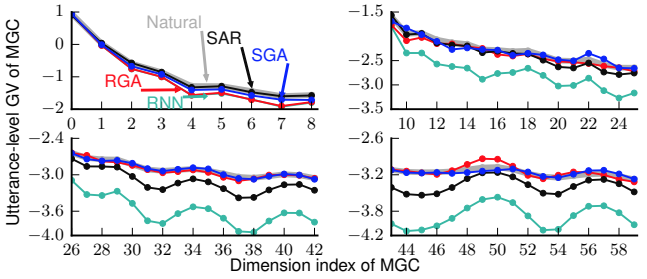

Fig. 5: Global variance (GV) of natural and generated MGC averaged on test set.

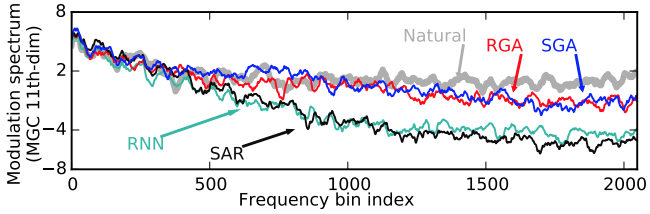

Fig. 6: Modulation spectrum of 11th dimension of MGC for test utterance AOZORAR_03372_T01.

that the AR model alleviated the over-smoothing effect. The GAN-based postfilter in both SGA and RGA also reduced the impact of over-smoothing. Further, GAN not only enhanced GV but also compensated for the modulation spectrum (MS) of the generated MGC throughout the whole frequency band, which can be seen from Fig. 6. However, neither SGA nor RGA outperformed SAR even though SAR did not boost the MS in the high frequency bands. After listening to the samples, we found that, while the samples of SGA and RGA had better spectrum details, they contained more artifacts. This indicated that generators in SGA and RGA may not optimally learn the distribution of natural MGC. Future work will look into details on SGA and RGA.

On waveform generation methods, the comparison among SAR-Wa, SAR-Pr, SAR-Pm, SAR-Wo, and vocoded speech showed that SAR-Wa significantly outperformed other waveform generation methods in terms of speech quality. More interestingly, SAR-Wa's quality was even judged to be better than other synthetic methods by using a 48 kHz sampling rate. The quality of SAR-Wa was also close to the 16 kHz vocoded samples from Abs-Pm and Abs-Wo. A closes inspection of the samples showed that the Wavenet vocoder in SAR-Wa had fewer artifacts, such as amplitude modulation in the waveform and other effects possibly due to conventional vocoders. However, the gap between SAR-Wa and Abs-Wo/Abs-Pm was still large in terms of speaker similarity. One reason may be that, while the Wavenet vocoder trained by using natural acoustic features may well learn the mapping from acoustic features to waveforms, during waveform generation it cannot compensate for the degradation of generated acoustic features caused by the imperfect acoustic model SAR. Note that this problem may not be resolved by training the Wavenet vocoder using generated acoustic features if the acoustic model is not good enough.

Among other waveform generation methods, the PML vocoder, while being similar to WORLD for analysis-by-synthesis, it lagged behind when using the generated acoustic features. This result is somewhat different from that in [11], and future work will investigate the reasons for this. Comparison between SAR-Pr and SAR-Wo indicated that the phase recovery method did not improve the quality or similarity score. We suspect that this is because the iterative process in the Griffin-Lim algorithm may introduce some artifacts that degraded speech quality.

4. CONCLUSION

This work was our initial step in building a framework in which recent acoustic modeling and waveform generation methods could be compared on a common ground by using a large-scale perceptual evaluation. This work only considered a few methods that could easily be plugged into the common TTS pipeline. On acoustic models, the results showed that the autoregressive model SAR could achieve a better performance than a normal RNN. This SAR was also easier to train than a GAN-based model. On waveform generation methods, this work demonstrated the potential of the Wavenet vocoder and the advantage of statistical waveform modeling compared with conventional deterministic approaches.

We intend to investigate the reasons for the experiments' results in more detail in future work. Meanwhile, we recently validated that SAR is a simple case of using normalizing flow [27] to transform the density of target data. We will try to improve SAR and the overall performance of speech synthesis system. Complex-valued models that support the modeling of waveform phases may also be included.

5. REFERENCES

[1] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[2] Jose Sotelo, Soroush Mehri, Kundan Kumar, João Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR (Workshop Track)*, 2017.

[3] 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.

[4] Felipe Espic, Cassia Valentini Botinhao, and Simon King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, 2017, pp. 1383–1387.

[5] Shinji Takaki, Hirokazu Kameoka, and Junichi Yamagishi, "Direct modeling of frequency spectra and waveform generation based on phase recovery for DNN-based speech synthesis," in *Proc. Interspeech*, 2017, pp. 1128–1132.

[6] Srikanth Ronanki, Oliver Watts, and Simon King, "A hierarchical encoder-decoder model for statistical parametric speech synthesis," in *Proc. Interspeech*, 2017, pp. 1133–1137.

[7] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, 2017, pp. 4910–4914.

[8] Xin Wang, Shinji Takaki, and Junichi Yamagishi, "An autoregressive recurrent mixture density network for parametric speech synthesis," in *Proc. ICASSP*, 2017, pp. 4895–4899.

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

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

[11] Gilles Degottex, Pierre Lanchantin, and Mark Gales, "A log domain pulse model for parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[12] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. ASSP*, vol. 32, no. 2, pp. 236–243, 1984.

[13] Xin Wang, Shinji Takaki, and Junichi Yamagishi, "An RNN-based quantized F0 model with multi-tier feedback links for text-to-speech synthesis," in *Proc. Interspeech*, 2017, pp. 1059–1063.

[14] Tokuda Keiichi, Yoshimura Takayoshi, Masuko Takashi, Kobayashi Takao, and Kitamura Tadashi, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 936–939.

[15] Ranniery Maia, Masami Akamine, and Mark J.F. Gales, "Complex cepstrum as phase information in statistical parametric speech synthesis," in *Proc. ICASSP*, 2012, pp. 4581–4584.

[16] Gilles Degottex and Daniel Erro, "A uniform phase representation for the harmonic model in speech synthesis applications," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, pp. 38, Oct 2014.

[17] Qiong Hu, Junichi Yamagishi, Korin Richmond, Katrick Subramanian, and Yannis Stylianou, "Initial investigation of speech synthesis based on complex-valued neural networks," in *Proc. ICASSP*, March 2016, pp. 5630–5634.

[18] Toru Nakashika, Shinji Takaki, and Junichi Yamagishi, "Complex-valued restricted Boltzmann machine for direct learning of frequency spectra," in *Proc. Interspeech*, 2017, pp. 4021–4025.

[19] 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, pp. 187–207, 1999.

[20] Takuhiro Kaneko, Shinji Takaki, Hirokazu Kameoka, and Junichi Yamagishi, "Generative adversarial network-based postfilter for STFT spectrograms," in *Proc. Interspeech*, 2017, pp. 3389–3393.

[21] Yajun Hu, Chuang Ding, Li Juan Liu, Zhen Hua Ling, and Li Rong Dai, "The USTC system for Blizzard Challenge 2017," in *Proc. Blizzard Challenge Workshop*, 2017.

[22] Boualem Boashash, "Estimating and interpreting the instantaneous frequency of a signal. ii. algorithms and applications," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 540–568, 1992.

[23] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. ICML*, 2017, pp. 1068–1077.

[24] Hisashi Kawai, Tomoki Toda, Jinfu Ni, Minoru Tsuzaki, and Keiichi Tokuda, "XIMERA: A new TTS from ATR based on corpus-based technologies," in *Proc. SSW5*, 2004, pp. 179–184.

[25] The HTS Working Group, "The Japanese TTS System 'Open JTalk'," 2015.

[26] Felix Weninger, Johannes Bergmann, and Björn Schuller, "Introducing CURRENT: The Munich open-source CUDA recurrent neural network toolkit," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 547–551, 2015.

[27] Danilo Rezende and Shakir Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, 2015, pp. 1530–1538.