INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Joint optimization of neural acoustic beamforming and dereverberation with x-vectors for robust speaker verification

Joon-Young Yang, Joon-Hyuk Chang

Department of Electronics and Computer Engineering
Hanyang University, Seoul, Republic of Korea
dreadbird@hanyang.ac.kr, jchang@hanyang.ac.kr

Abstract

In this paper, we investigate the deep neural network (DNN) supported acoustic beamforming and dereverberation as the front-end of the x-vector speaker verification (SV) framework in a noisy and reverberant environment. Firstly, a DNN for supporting either the classical beamforming (e. g. MVDR) or the dereverberation (e. g. WPE) algorithm is trained on multi-channel speech signals. Next, an x-vector speaker embedding network is trained on top of the enhanced speech features to classify the training speakers. Finally, after the separate training stages are over, either one or both of the DNN supported beamforming and dereverberation modules are serially connected to the x-vector network, and jointly trained to optimize the common objective of speaker classification. Experiments on the artificially generated speech dataset using simulated and real room impulse responses (RIRs) with various types of domestic noise samples show that jointly training the supportive neural network models along with the x-vector network within the classical speech enhancement framework brings significant performance gain for robust text-independent (TI) SV.

Index Terms: acoustic beamforming, dereverberation, deep neural network, speaker verification, joint training.

1. Introduction

For the past decade, deep neural networks (DNNs) have been widely adopted for various speech processing applications. Two of such extensions are neural network supported beamforming [1] and dereverberation [2], where the DNNs replace the conventional statistical model-based signal statistics estimation routines in a trainable, data-driven fashion, while retaining the theoretically well-driven classical speech enhancement algorithms. Perhaps the most successful use of such applications would be as the front-ends for robust multi-channel automatic speech recognition (ASR), where the minimum variance distortionless response (MVDR) [3] beamformer and the weighted prediction error (WPE) [4] dereverberation algorithms supported by the DNNs were proven to be effective [1, 2, 5, 6, 7] when evaluated on the dataset collected for the past CHiME-3 [8] or REVERB [9] challenges. Furthermore, several studies [5, 7, 10] have shown that the joint optimization of the front-end DNNs involved with the classical beamforming/dereverberation operations and the back-end neural network for ASR can result in additional performance improvement.

On the other hand, in the robust text-independent (TI) speaker verification (SV) area, only a few studies have investigated the efficacy of the beamforming or dereverberation algorithms against the multi-channel speech signal generated in a noisy or reverberant environment [11, 12]. Specifically, in [11, 12], the combinations of various denoising and dereverberating front-ends have been extensively examined for the TI SV task using the real multi-channel speech dataset rerecorded in a reverberant room. However, their speaker embedding model is the i-vector [13], which is known to lack the ability to model multi-style data and hence the robustness, compared to the recently proposed deep speaker embedding model [14, 15]. Furthermore, their systems are not subject to joint optimization due to the different nature of the front-end and the back-end.

In this paper, we investigate the utility of the DNN supported MVDR beamformer and the WPE dereverberation jointly optimized with the x-vector [14] deep speaker embedding. One major difference from the previous studies is that, in [11, 12], the i-vector extractor is shared among the different pre-processing front-ends so that the performance of such front-ends can be compared using a fixed back-end. However, we build speaker embedding models separately from each output stream of the different front-ends, which allows us to investigate the appropriateness of each front-end and the corresponding enhanced signal for deep speaker modeling itself. Moreover, motivated by the recent studies [6, 11, 12] that have shown the effectiveness of the cascade structure of WPE dereverberation followed by neural beamforming for robust speech and speaker recognition tasks, we also examine the cascade of the DNN supported WPE, DNN supported MVDR, and the x-vector network, and further jointly train the whole system under the final objective of speaker classification.

2. System overview

2.1. Signal model

Assuming that the speech signal is collected in a noisy and reverberant room using D microphones, the observed signal comprises an additive mixture of the reverberated speech source and noise source signals. In the short-time Fourier transform (STFT) domain, the signal model can be represented as follows:

$$\mathbf{y}_{t,f} = \mathbf{x}_{t,f} + \mathbf{n}_{t,f} = \mathbf{x}_{t,f}^{(\text{early})} + \mathbf{x}_{t,f}^{(\text{late})} + \mathbf{n}_{t,f}^{(\text{early})} + \mathbf{n}_{t,f}^{(\text{late})} \quad (1)$$

where $\mathbf{y}_{t,f}$ is the D-dimensional observation vector, $\mathbf{x}_{t,f}$ and $\mathbf{n}_{t,f}$ are the speech and noise source signals convolved with the room impulse responses (RIRs) measured for each of the pairs of the receiver and source positions, and the superscripts (early) and (late) denote the direct plus early reflection and the late reverberation signal components, respectively. Note that, in this work, we consider the direct plus early reflection components to be obtained using the first 50 ms after the main peak of the RIR while the remaining part contributes to the undesired late reverberation.

2.2. DNN supported MVDR beamforming

Classical MVDR beamformer is designed to minimize the residual noise while constraining the output speech to be dis-

http://dx.doi.org/10.21437/Interspeech.2019-1356

tortionless [3]. Solving the minimization problem gives the MVDR gain for each frequency as below:

$$\mathbf{w}_{\text{MVDR}} = \frac{\mathbf{\Phi}_{nn}^{-1}\mathbf{\Phi}_{xx}}{\text{tr}(\mathbf{\Phi}_{nn}^{-1}\mathbf{\Phi}_{xx})}\mathbf{u}_{\text{ref}} \tag{2}$$

where $\mathbf{\Phi}_{xx}$ and $\mathbf{\Phi}_{nn}$ respectively denote the power spectral density (PSD) matrices of the speech and noise components, and $\mathbf{u}_{\text{ref}}$ is a one-hot vector for reference channel selection [3]. Note that the frequency index f is omitted for better readability. The beamformed output is obtained by multiplying the gain to the observation:

$$\hat{\mathbf{x}}_{t,f} = \mathbf{w}_{\text{MVDR}}^{\text{H}}\mathbf{y}_{t,f}. \tag{3}$$

Neural network supported mask-based beamforming [1] exploits the DNN for PSD matrix estimation by estimating the spectral masks for speech and noise in a channel-independent manner. In order to generate masks within the range of values between 0 and 1, the output layers of the DNN comprise the sigmoid units, and the training targets are given as the ideal binary masks (IBMs) formulated as follows:

$$M_{d,t,f}^{(\nu)} = \begin{cases} 1, & \frac{|x_{d,t,f}^{(\text{early})}|^2}{|x_{d,t,f}^{(\text{late})}+n_{d,t,f}|^2} \gtrless_{n}^{x} \theta_f^{(\nu)} \\ 0, & \text{otherwise} \end{cases}, \quad \nu \in \{x, n\} \tag{4}$$

where ν denotes the signal attribute, d is the microphone channel index, and θ_f is a predefined decision threshold[1] [1], respectively. Note that the above mask formulation expects the beamformer to also have a dereverberation effect to some extent, as mentioned in the previous study [6]. In the inference stage, the soft masks are separately obtained for each microphone, and then averaged to be commonly applied to all channels when calculating the PSD matrices as follows:

$$\mathbf{\Phi}_{\nu\nu} = \sum_t \hat{M}_{t,f}^{(\nu)}\mathbf{y}_{t,f}\mathbf{y}_{t,f}^{\text{H}} \Big/ \sum_t \hat{M}_{t,f}^{(\nu)}, \quad \nu \in \{x, n\} \tag{5}$$

where $\hat{M}_{t,f}$ denotes the average of the soft spectral masks estimated from the DNN. During the training, the binary cross-entropy loss between the estimated and the target masks are used as the loss function.

2.3. DNN supported WPE dereverberation

Classical WPE algorithm finds the linear prediction (LP) filter to estimate the late reverberation and subtract it from the observation to obtain the maximum likelihood (ML) estimate of the early arriving speech which is assumed to be sampled from a complex normal distribution with zero mean and time-varying variance $\lambda_{t,f}$ [4]. Since no closed form solution of the LP filter coefficients for the above ML optimization problem exists, borrowing some notations from [6], an iterative procedure for estimating the filter coefficients is given as follows:

Step 1) $$\lambda_{t,f} = \frac{1}{D}\sum_d |\hat{x}_{d,t,f}^{(\text{early})}|^2 \tag{6}$$

Step 2) $$\mathbf{R}_f = \sum_t \frac{\tilde{\mathbf{y}}_{t-\Delta,f}\tilde{\mathbf{y}}_{t-\Delta,f}^{\text{H}}}{\lambda_{t,f}} \in \mathbb{C}^{DK\times DK} \tag{7}$$

$$\mathbf{P}_f = \sum_t \frac{\tilde{\mathbf{y}}_{t-\Delta,f}\mathbf{y}_{t,f}^{\text{H}}}{\lambda_{t,f}} \in \mathbb{C}^{DK\times D} \tag{8}$$

$$\mathbf{G}_f = \mathbf{R}_f^{-1}\mathbf{P}_f \in \mathbb{C}^{DK\times D} \tag{9}$$

Step 3) $$\hat{\mathbf{x}}_{t,f}^{(\text{early})} = \mathbf{y}_{t,f} - \mathbf{G}_f^{\text{H}}\tilde{\mathbf{y}}_{t-\Delta,f} \tag{10}$$

where $\lambda_{t,f}$ is the average power of the estimated early arriving speech, K is the order of the LP filter, Δ is a delay for LP, and $\tilde{\mathbf{y}}_{t-\Delta,f}$ and $\mathbf{G}_f$ are the stacked representations (from Δ-th to $(\Delta + K - 1)$-th past time frames) of the observation and the filter coefficients, respectively.

Neural network supported WPE[2] [2] simply replaces the power estimation routine in Eq. (6) with a DNN, free from iterations. Specifically, in this work, the DNN is trained to estimate the log-scale power spectra (LPS) of $x_{d,t,f}^{(\text{early})} + n_{d,t,f}^{(\text{early})}$ given the LPS of $x_{d,t,f} + n_{d,t,f}$ as the input, expecting the network to remove the late reverberation components from the reverberated point-source signals. After the training is completed, the output LPS are transformed back to linear-scale, and then put to Eq. (6) to obtain the average power. The loss function is set to the mean squared error (MSE) between the estimated and the target LPS.

2.4. X-vector speaker embedding

X-vector speaker embedding network is fed with a variable-length sequence of frame-level acoustic features, such as mel-filterbank energies (MFBEs). A stack of time-delay neural network (TDNN) layers extracts the context-dependent speaker information within the temporal receptive field of 15 consecutive input frames, which are then aggregated to the utterance-level mean and standard deviation vectors by the following statistics pooling layer [14]. Finally, the pooled statistics are passed through the two fully-connected layers, and the network is trained to classify the training speakers' identity at the softmax output layer under the supervision of cross-entropy loss. After the training is finished, the pre-activation of the first fully-connected layer is used as the speaker embedding.

2.5. Joint training

As shown in the previous studies [5, 7, 10], neural beamforming architectures followed by a back-end DNN for classification have potential room for performance improvement through joint optimization of the whole system based on the final classification criterion. Since the classical beamforming operations supported by a neural network are fully differentiable, it allows the gradients from the back-end loss to be backpropagated through the whole system [5]. Similarly, neural network supported WPE is also subject to joint optimization, but the studies regarding the joint training of the DNN supported WPE are still missing. Therefore, in this work, we propose to jointly optimize the DNN supported WPE with the back-end speaker embedding network using the final cross-entropy loss for speaker classification. Furthermore, in the recent study [6], a cascaded structure of WPE followed by neural beamforming has been proven effective for robust speech recognition task. Motivated by that, we also perform joint training for the stacked architecture of neural WPE, neural beamforming, and deep speaker embedding. Our overall system configuration is depicted in Fig. 1.

The complex-valued operations needed for joint optimization are implemented simply by splitting the real and imaginary parts and separately performing real-valued operations. For the complex-valued matrix inverse operations required in Eq. (2) and Eq. (9), the following formula would suffice [5, 16]:

$$\Re(\mathbf{C}^{-1}) = (\mathbf{A} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B})^{-1} \tag{11}$$

$$\Im(\mathbf{C}^{-1}) = -(\mathbf{A} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}\mathbf{A}^{-1} \tag{12}$$

[1] The implementation is based on https://github.com/fgnt/nn-gev

[2] The implementation is based on https://github.com/fgnt/nara_wpe

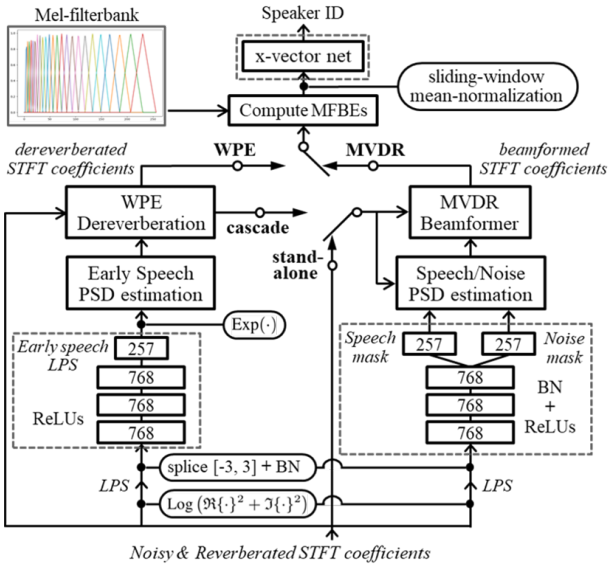

Figure 1: *Overall system configuration.*

where $\mathbf{C} = \mathbf{A} + i\mathbf{B}$ is a complex-valued matrix, and $\mathbf{A}$ and $\mathbf{B}$ are invertible real-valued matrices.

3. Experimental setup

3.1. Dataset

All the experiments were conducted on a large vocabulary continuous Korean speech dataset developed by the Speech Information Technology and Industry Promotion Center (SiTEC) [17, 18]. The dataset contains phonetically balanced, single-channel clean-room speech recordings sampled at 16 kHz, where each speaker contains 103 to 105 utterances of average duration of 5.18 seconds. We split the dataset into two parts with equally distributed genders, where the training set contains 84,103 utterances from 800 speakers, and the evaluation set comprises 200 speakers with 25 utterances per speaker.

In order to generate multi-channel speech data corrupted with noise and reverberation, we used the RIR generator [19] to simulate various room environments. We prepared 5,000 and 1,000 room configurations for training and evaluation, respectively, according to the randomized settings in Table 1. For each room, 4 different random sets of RIRs are generated, where each set contains a single RIR for a speech source and 1 to 3 RIRs for noise sources, and both the reverberated speech with and without noise addition are used for training, which basically doubles the size of the training dataset. For the noise addition, the audio samples from the domestic audio tagging dataset of the detection and classification of acoustic scenes and events (DCASE) 2016 Challenge [20] were artificially added to the clean speech at various SNR levels from 0 dB to 20 dB. The entire set of 6,137 samples was randomly split into two sets containing 4,091 and 2,046 samples for training and evaluation, respectively, each of which comprises various types of domestic noises such as child/adult speech, video game, TV, percussive sounds, etc. We corrupted each utterance with randomly selected RIRs and noise samples, and made two copies of the entire training set. Note that, due to the huge resource requirement for processing mini-batched multi-channel utterances, especially for the joint training, we opt to perform all our experiments on 2-channel audio.

Table 1: *Parameters for random RIR generation.*

Parameter		Value
Room size	small	$[4\times4\times2]$ m^3 to $[10\times10\times5]$ m^3
	medium	$[10\times10\times2]$ m^3 to $[30\times30\times5]$ m^3
Microphone array		2 mics with 9.5 cm of spacing
Source-Receiver dist.		0 m to 4 m
RT60		0.3 sec to 0.8 sec

In addition to the simulated RIRs for evaluation, we also used some real RIRs taken from the REVERB Challenge 2014 dataset [9], where only the RIRs from the microphone 1 and 5 of the circular array are used.

3.2. Model specifications

Since our main purpose is to investigate and verify the utility of the joint optimization framework of the DNN supported MVDR/WPE and the deep speaker embeddings, we chose to use rather simple fully-connected neural networks (FCNN) for the front-ends. Both the DNNs for mask estimation and LPS estimation are fed by 257-dimensional LPS features spliced with 3 left and 3 right context frames, apply batch-normalization (BN) [21] to the inputs, and consist of 3 hidden layers of 768 rectified linear units (ReLUs). Two output layers of 257 sigmoid units were constructed for estimating the speech and noise masks, whereas one output layer of 257 linear units for estimating the LPS of the early reflection signal. Note that both the MVDR and WPE work in batch-mode, and the parameters of the LP filter for the WPE were fixed to $(\Delta, K) = (3, 10)$.

For the x-vector model, we used the exact same model structure in [14]. The input features were 24-dimensional MFBEs mean-normalized within a sliding window of up to 3 seconds, and the output layer consists of 800 units which is equal to the number of the training speakers. Our baseline model is the x-vector trained on the unprocessed observation, where only the first channel is used for training and evaluation. Similarly, for the WPE front-ends, only the first channel output was fed to the x-vector network.

3.3. Training

In order to perform joint training in an end-to-end fashion, we opt to unify some experimental settings for training the networks. Firstly, the frame and hop size for acoustic feature extraction were set to 32 ms and 8 ms in order to remove the need for transforming back to the time-domain. Secondly, we unified the mini-batching scheme by following that of the x-vector model. Specifically, a single mini-batch was composed by gathering randomly cropped utterances of variable duration within 3 to 8 seconds for training the networks, while the whole utterance is fed to the networks in the inference stage. Finally, the initial and the final learning rate were set equal for all three models.

All the networks were trained using the Adam [22] algorithm, where the FCNNs were trained for 40 epochs with the mini-batch size of 32 and the x-vector model was trained for 60 epochs with the mini-batch size of 64. The initial learning rate was set to 10^{-3} and annealed three times, each time by $1/3$ at the predefined epochs. Dropout of 20% and *l2*-regularization of the weights were applied to regularize the training. For the joint

Table 2: *EER (%) on the simulated and the real RIRs.*

Model	*simulated*	*real*
Unprocessed	5.325	5.702
MVDR_{DNN}	4.315	5.414
$\text{MVDR}_{\text{oracle}}$	3.721	4.707
MVDR_{DNN} + JT	3.955	4.892
WPE_{DNN}	4.686	4.749
$\text{WPE}_{\text{oracle}}$	4.807	4.769
$\text{WPE}_{\text{iterative}}$	4.746	4.812
WPE_{DNN} + JT	4.182	4.526
WPE_{DNN} + MVDR_{DNN}	3.733	4.538
$\text{WPE}_{\text{oracle}}$ + $\text{MVDR}_{\text{oracle}}$	3.221	3.819
WPE_{DNN} + MVDR_{DNN} + JT	3.297	3.967

training, the number of training epochs was set to 20, the initial learning rate was fixed to the final learning rate used for the separate training, and the mini-batch size was set to 64, except for the case of jointly training the cascade of all three models in which we set the mini-batch size to 52 due to the GPU memory limitation. All the networks were implemented in TensorFlow [23] and trained using a single or two NVIDIA GTX 1080 Ti GPUs.

For the speaker verification back-end, the training set x-vectors were centered, dimensionality reduced to 200 by LDA, and length-normalized [24] to train the two-covariance [25] model.

3.4. Evaluation

For the evaluation, we created 60,000 target and 221,850 non-target trials from the evluation set without including any cross-gender trials. The speaker verification results were evaluated in terms of the equal error rate (EER).

4. Results and analysis

4.1. Results on simulated RIRs

Table 2 shows the speaker verification results on the simluated RIRs. In the table, the subcript *oracle* denotes the DNN supported MVDR (WPE) provided with the oracle binary masks (early arriving LPS), *iterative* denotes the conventional iterative WPE, and JT denotes the joint training. For the MVDR front-ends, the DNN MVDR and the oracle MVDR showed 19.0% and 30.1% of releative improvement to the unprocessed model, while the gap was reduce by the joint training which brought an additional improvement of 6.7%. On the other hand, the DNN WPE outperformed the unprocessed by 12%, and also slightly outperformed both the oracle and the iterative WPE. We conjecture that the main reason for the performance gap between the oracle and the DNN WPE is the empirically chosen LP parameters for WPE which may not be optimal, or that the oracle LPS described in Section 2.3 may be not optimal but appropriate enough as the training targets of the DNN when the DNN supported WPE is adopted as the front-end of the x-vector for TI SV. Overall, the jointly trained DNN WPE outperformed all of the separately trained, oracle, and iterative WPE front-ends by considerable margin.

Comparing the results of the stand-alone MVDR and the WPE, the MVDR outperformed the WPE possibly because the latter only dereverberates whereas the former dereverberates and denoises simultaneously, but it is clearly seen that the classical WPE algorithm is much more insensitive to the estimation errors caused by the DNN than the MVDR. The performance of the cascaded WPE and MVDR front-ends are presented in the last three rows of the table. As shown in the table, both the oracle and the DNN supported cascades show significant improvement compared to their stand-alone counterparts even though the spectral masks designed to denoise and dereverberate the observed signal are directly applied to the dereverberated observations. Meanwhile, the performance gap between the oracle and the DNN supported cascade seems to be directly delivered from that of the stand-alone MVDR front-end. This performance gap was overcome up to 85.2% by jointly optimizing all of the constituent models with respect to the final classification loss.

4.2. Results on real RIRs

Since the results on the real RIRs show similar patterns to those on the simulated RIRs, we rather focus on the analysis of the performance change between the two conditions. Firstly, observing the MVDR front-ends, the performances on the simulated RIRs were degraded about 23.7%–26.5% for all three models when evaluated on the real RIRs, where the main cause seems to be the mismatch between the real and the simulated room environment. On the other hand, the WPE front-ends were still robust to the mismatch, showing 1.3% and 1.4% of performance degradation for the DNN supported and the iterative models, respectively. For the jointly trained WPE, the relative performance degradation was 8.2%, which was relatively large compared to the other stand-alone WPE front-ends. One possible explanation is that the back-end is jointly optimized with the front-end to fit to the simulated room, thus enlarging the mismatch between the two conditions. For the oracle WPE, even a slight improvement was observed, which might be because the number of real RIRs used in the experiment is too small. Nonetheless, the jointly trained WPE showed the best performance among the stand-alone WPE front-ends, and further outperformed the stand-alone MVDR front-ends due to the smaller performance degradation. Finally, the cascade of WPE and MVDR showed 18.6%–21.6% of degradation, which seems to be propagated mainly from that of the MVDR front-end. Similar to the results on the simulated room environment, the performance gap between the oracle and the DNN supported cascade was reduced up to 79.4% after the joint training.

5. Conclusions

In this paper, we investigated the effectiveness of the neural network supported MVDR and WPE front-ends jointly trained with the deep speaker embedding back-end, including their stand-alone and the cascaded use cases. While the stand-alone MVDR front-ends outperformed the WPE on the simulated RIR setup, the latter showed similar or better performance on the real RIRs, revealing great robustness to the environmental mismatch. Furthermore, joint optimization of the whole system was proven to be effective, bringing significant performance improvement of the separately trained modules.

6. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00474, Intelligent Signal Processing for AI Speaker Voice Guardian).

7. References

[1] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 196–200.

[2] K. Kinoshita, M. Delcroix, H. Kwon, T. Hori, and T. Nakatani, "Neural network based spectrum estimation for online WPE dereverberation," in *Proc. Interspeech*, 2017, pp. 384–388.

[3] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 18, no. 2, pp. 260–276, 2010.

[4] Y. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[5] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1274–1288, 2017.

[6] L. Drude, C. Boeddeker, J. Heymann, R. Haeb-Umbach, K. Kinoshita, M. Delcroix, and T. Nakatani, "Integrating neural network based beamforming and weighted prediction error dereverberation," in *Proc. Interspeech*, 2018, pp. 3043–3047.

[7] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel ASR system," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 196–200.

[8] J. Barker, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Automatic Speech Recognition Understanding*, 2015, pp. 504–511.

[9] K. Kinoshita, M. Delcroix, T. Yoshioka, and T. Nakatani, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[10] X. Xiao, C. Xu, Z. Zhang, S. Zhao, S. Sun, S. Watanabe, L. Wang, L. Xie, D. L. Jones, E. S. Chng *et al.*, "A study of learning based beamforming methods for speech recognition," in *CHiME 2016 workshop*, 2016, pp. 26–31.

[11] L. Mosner, P. Matejka, O. Novotny, and J. H. Cernocky, "Dereverberation and beamforming in far-field speaker recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5254–5258.

[12] ——, "Dereverberation and beamforming in robust far-field speaker recognition," in *Proc. Interspeech*, 2018, pp. 1334–1338.

[13] N. Dehak, P. Kenny, J. Dehak, R. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5329–5333.

[15] O. Novotny, O. Plchot, P. Matejka, L. Mosner, and O. Glembek, "On the use of x-vectors for robust speaker recognition," in *Proc. Odyssey*, 2018, pp. 168–175.

[16] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," 2008, *Technical University of Denmark*.

[17] J. Jeon, S. Wee, and M. Chung, "Generating pronunciation dictionary by analyzing phonological variations frequently found in spoken korean," in *Proc. International Conference on Speech Processing*, 1997, pp. 519–524.

[18] J. Jeon, S. Cha, M. Chung, and J. Park, "Automatic generation of korean pronunciation variants by multistage applications of phonological rules," in *Proc. International Conference on Spoken Language Processing*, 1998, pp. 1943–1946.

[19] E. A. P. Habets, "Room impulse response generator," 2010. [Online]. Available: http://home.tiscali.nl/ehabets/rir_generator.html

[20] [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-audio-tagging

[21] S. Ioffe and S. Christian, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv: 1502.03167*.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv: 1412.6980*.

[23] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[24] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[25] N. Brummer and E. D. Villiers, "The speaker partitioning problem," in *Proc. Odyssey*, 2010, pp. 194–201.