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

Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system

Chanwoo Kim, Minkyu Shin, Abhinav Garg, and Dhananjaya Gowda

Samsung Research, Seoul, South Korea
{chanw.com, mk0211.shin, abhinav.garg, d.gowda}@samsung.com

Abstract

In this paper, we present an improved vocal tract length perturbation (VTLP) algorithm as a data augmentation technique. VTLP is usually accomplished by adjusting the center frequencies of mel filterbank in [1]. Compared to the conventional approach, we re-synthesize waveforms from the frequency-warped spectra using overlap and addition (OLA). This approach had two advantages: First, we can apply an "acoustic simulator" [2, 3] after performing the VTLP-based frequency warping. Second, we may use a different window length for frequency warping from that used in feature processing. We observe that the best performance was obtained when the warping coefficient distribution is between 0.8 and 1.2, and the window length is 50 ms. We obtained 3.66 % WER and 12.39 % WER on the Librispeech test-clean and test-other using an attention-based end-to-end speech recognition system without using any Language Models (LMs). Using the shallow-fusion technique with a Transformer LM, we achieved 2.44% WER and 8.29 % WER on the Librispeech test-clean and test-other sets. To the best of our knowledge, the 2.44 % WER on the test-clean is the best result ever reported on this test set.

Index Terms: Data augmentation, vocal tract length perturbation, vocal tract length normalization, end-to-end speech recognition

1. Introduction

Recently, deep learning technology has greatly enhanced speech recognition accuracy [4, 5, 6, 7]. Thanks to these advances, voice assistant devices such as Google Home [2, 8] or Amazon Alexa are actively used at many homes. To build high performance speech recognition systems, we need to have a large amount of data covering various domains. In [9], it has been shown that we need sufficiently large training set to achieve high speech recognition accuracy for difficult tasks like video captioning.

For far-field speech recognition, the impact of noise and reverberation is much larger than near-field cases. Traditional approaches to far-field speech recognition include noise robust feature extraction algorithms [10, 11], multi-microphone approaches [12, 13, 14], and approaches utilizing the "precedence effect" [15, 16]. Another popular approach for enhancing the robustness is using data augmentation [17, 3, 18]. We have been using the "acoustic simulator" [2, 3] to generate simulated utterances in millions of different room dimensions, a wide distribution of reverberation time and signal-to-noise ratios.

In a similar spirit, to tackle the speaker variability issue, Vocal Tract Length Perturbation (VTLP) has been proposed [1]. The Vocal Tract Length Perturbation (VTLP) algorithm is motivated based on the Vocal Tract Length Normalization (VTLN) algorithm. There are many factors which make speech from different speakers have different acoustic characteristics. It has been agreed that one of such factors is the Vocal Tract Length (VTL). The variations in VTL may be modeled as frequency warping [19]. In VTLN, the objective is to find the best frequency warping factor α to maximize the likelihood of feature given the model parameters. In the VTLN approach, frequency warping is performed both in training and evaluation phases to maximize the data likelihood. The objective of VTLN is to normalize the effect of spectral variation due to variations in VTL of different speakers.

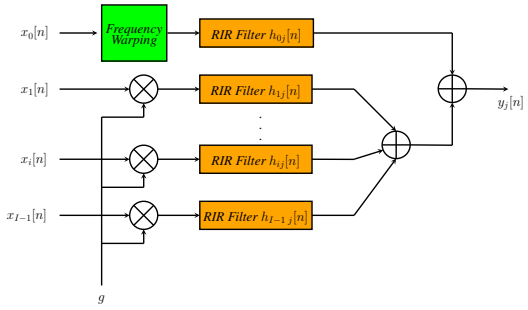

Figure 1: *The entire structure used for data augmentation.*

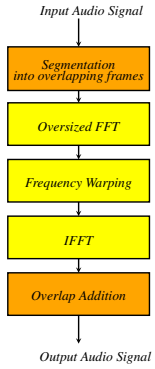

Figure 2: *A pipeline for performing Vocal Tract Length Perturbation (VTLP).*

On the contrary, VTLP tries to add more data to the training set by applying various frequency warping factors representing different VTLs. Conventionally, VTLP and VTLN have been usually performed by adjusting the center frequencies of mel filterbank employed for feature extraction [20, 1]. This conventional approach has the advantage of not requiring an additional step of resynthesizing speech waveform from the frequency-warped spectra. However, in the proposed system, we perform the frequency warping on the spectrum and resynthesize speech using Inverse Fast Fourier Transform (IFFT) and OverLap Add (OLA). This approach had two advantages. First, it is the correct way of performing data augmentation using both the Room Impulse Response (RIR) filter and VTLP.

http://dx.doi.org/10.21437/Interspeech.2019-3227

The entire structure for our data augmentation is depicted in Fig. 1. Assuming that there are I sound sources and J microphones, the received signal at microphone j is given by:

$$y_j[n] = \mathcal{F}\left(x_0[n]|\,\alpha\right) * h_{0j}[n] + g\sum_{i=1}^{I-1} x_i[n] * h_{ij}[n], \quad (1)$$

where $x_0[n]$ is the target speech signal, and $x_i[n], 1 \le i \le I-1$ are the noise signals. $h_{ij}[n]$ is the room impulse response related to the locations of the i-th sound source and j-th microphone. g is the gain factor adjusting the relative intensity of noise signals with respect to that of the target speech signal $x_0[n]$. $\mathcal{F}(\cdot|\alpha)$ is the frequency warping transform and α is the warping factor which will be described in Sec. 2. Since the frequency warping is a non-linear operation, the order of applying frequency warping and RIR filters affects the final resynthesized speech. Since the objective of VTLP is simulating the speaker variability due to the difference in vocal tract length, the RIR filter should be applied after the frequency warping. If warping is performed during feature extraction, then the net effect on the first term on the right hand becomes $\mathcal{F}\left(x_0[n] * h_{0j}[n]|\alpha\right)$, which is different from the original intention of the system.

Second, it has been frequently observed that there is time-frequency resolution trade-off in speech processing [21, 22, 23]. Longer-windows usually provide a better frequency resolution and have been shown to be more effective in power distribution normalization [21] and noise suppression [24]. As will be described in Sec. 4, for the VTLP processing described in this paper, we observe that better performance is achieved when the window length is 50 *m*s, which is longer than the 25 *m*s window length we used for feature processing.

The rest of the paper is organized as follows: We describe the VTLP processing in detail in Sec. 2. The end-to-end speech recognition system with data augmentation is described in Sec. 3. Experimental results that demonstrates the effectiveness of the VTLP processing is presented in Sec. 4. We conclude in Sec. 5.

2. Data augmentation using Vocal Tract Length Perturbation (VTLP)

In this section, we will explain each component of the VTLP processing in detail. The entire structure of the VTLP algorithm is shown in Fig. 2.

2.1. Oversized FFT and frequency warping

The first step of the VTLP processing is to segment speech into successive frames by applying a 50-*ms* Hanning window. The choice of this rather long window length is based on our experimental results in Sec. 4. At 16-*kHz*, even though the minimum required FFT size that is power of two is $K = 1024$, we use a FFT size of $U = 16$ times of this value to have a better resolution while performing frequency warping. Note that when preforming a DFT of size K, K discrete frequencies are uniformly sampled between 0 and 2π. Thus the frequency resolution becomes $\frac{2\pi}{K}$.

For a specific frame index m, we obtain the frequency-warped short-time spectrum $Y_K[m, e^{j\omega_k}]$ whose FFT size is K from the original spectrum $X_{UK}[m, e^{jv_k}]$ whose FFT size is UK using the following relation:

$$\begin{aligned} Y_K[m, e^{j\omega_k}] &= X_{UK}[m, e^{jv_{k_0}}] \\ &= X_{UK}[m, e^{j\phi_\alpha(\omega_k)}], \quad 0 \le k \le \frac{K}{2}, \end{aligned} \quad (2)$$

where m is the frame index, and $\phi_\alpha(\omega_k)$ is the frequency warping function, which will be explained in Sec. 2.2. Discrete-time frequencies ω_k and v_{k_o} are defined by $\omega_k = \frac{2\pi k}{K}$ and $v_{k_o} = \frac{2\pi k_o}{UK}$, respectively where K is the FFT size, and U is the oversize FFT factor. As mentioned earlier, we use the U value of 16. We obtain the frequency-warped spectrum $Y(e^{j\omega_k})$ for the lower half discrete frequency region of $0 \le k \le \frac{K}{2}$, since the spectrum for the upper half frequency region may be obtained using the complex conjugate symmetry property [25].

In obtaining the warped spectrum $Y_K[m, e^{j\omega_k}]$ using (2), one issue is that $X_{UK}[m, e^{jv_k}]$ is defined only for discrete-values of k. Thus, $\phi_\alpha(\omega_k)$ in (2), should be able to be represented as v_{k_0} where k_0 is a certain integer. Since v_{k_0} is defined to be $\frac{2\pi k_0}{UK}$, we use the following equation to find k_o:

$$\frac{2\pi k_o}{UK} \approx \phi_\alpha(\omega_k). \quad (3)$$

Thus, we obtain:

$$k_o \approx \frac{UK\phi_\alpha(\omega_k)}{2\pi}. \quad (4)$$

From the above equation, the k_o value is given by rounding as shown below:

$$k_o = \left\lfloor \frac{UK\phi_\alpha(\omega_k)}{2\pi} + 0.5 \right\rfloor \quad (5)$$

where $\lfloor\cdot\rfloor$ is the floor operator, and 0.5 is added for the rounding purpose. From (3) and , we observe that the

Using (2) and (5) we obtain:

$$\begin{aligned} Y_K[m, e^{j\omega_k}] &= X_{UK}[e^{jv_{k_o}}] \\ &= X_{UK}\left[m, e^{jv\left\lfloor \frac{UK\phi_\alpha(\omega_k)}{2\pi}+0.5\right\rfloor}\right], \\ & \qquad 0 \le k \le \frac{K}{2}, \end{aligned} \quad (6)$$

where discrete-time frequencies ω_k and v_k are defined by $\omega_k = \frac{2\pi k}{K}$ and $v_k = \frac{2\pi k}{UK}$ as mentioned before.

2.2. Frequency warping based on bilinear transformation

In this section, we discuss the frequency warping function denoted by $\phi_\alpha(\omega_k)$ in (2) and (6). Several different warping equations have been proposed for VTLN [19]. The piecewise linear rule and the bilinear rule might be one of the widely used warping methods. The piecewise linear rule is the simplest way of performing frequency warping and this rule is represented by the following equation:

$$\omega'_k = \begin{cases} \omega\alpha, & \omega \le \omega_{hi}\frac{\min(\alpha,1)}{\alpha} \\ \pi - \frac{\pi-\omega_{hi}\min(\alpha,1)}{\pi-\omega_{hi}\frac{\min(\alpha,1)}{\alpha}}\left(\pi-\omega\right), & \text{otherwise} \end{cases} \quad (7)$$

where $\omega_k = \frac{2\pi k}{K}$ is the input discrete-time frequency and $\omega'_k = \frac{2\pi k'}{K}$ is the output discrete-time frequency. In the bilinear transformation, the relation between the input and output discrete-time frequencies is given by:

$$\omega'_k = \omega_k + 2\tan^{-1}\left(\frac{(1-\alpha)\sin(\omega_k)}{1-(1-\alpha)\cos(\omega_k)}\right). \quad (8)$$

where w_k is the discrete-time frequency defined by $w_k = \frac{2\pi k}{K}$, and K is the DFT size. Fig. 4 shows the relationship between

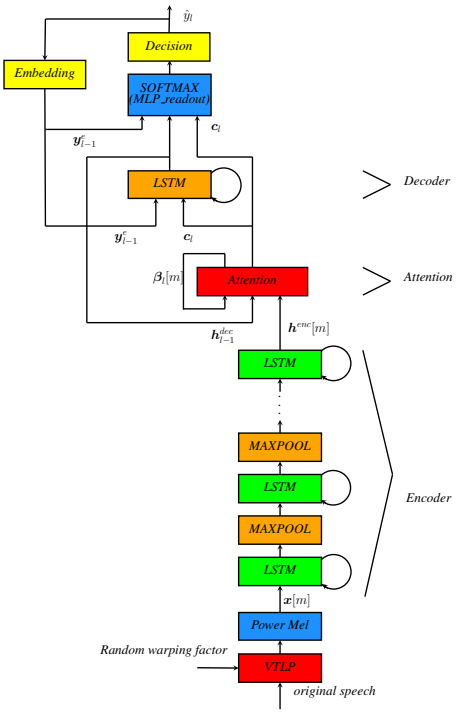

Figure 3: *The structure of the entire end-to-end speech recognition system.*

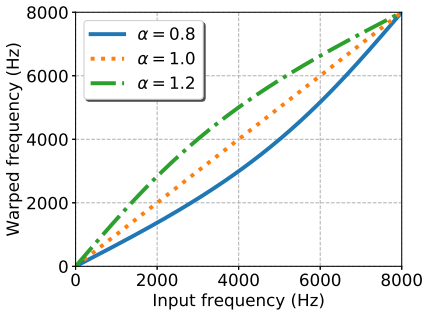

Figure 4: *The relation between the input frequency and the warped frequency using the bilinear transformation* (8).

the input and output frequencies. In our work, we use the bilinear rule in (8), since this approach has shown better performance than the piecewise linear rule in speech recognition experiments using VTLN [19].

3. End-to-end speech recognition with Vocal Tract Length Perturbation (VTLP)

We used the RETURNN speech recognition system [26, 27] with various modifications [7]. $\boldsymbol{x}[m]$ and $\hat{y}_l$ are the input power mel filterbank vector and the hypothesis output label, respectively. m is the input frame index and l is the decoder output step index. For the input feature, instead of the conventional Mel Filterbank Cepstral Coefficients (MFCCs) or log-mel, we use the power mel filterbank coefficients with the power nonlinearity of $(\cdot)^{\frac{1}{15}}$. This power law nonlinearty is motivated by our previous research in [10, 28, 24]. VTLP processing is performed using the system described in Sec. 2 with a uniformly random warping coefficient α between α_{min} and α_{max}. As will be explained in Tables 2 and 3, $\alpha_{min} = 0.8$ and $\alpha_{max} = 1.2$ were the best choices in our experiments. The warping coefficient is randomly generated for every training example. As a result, the acoustic model is trained using examples that are never repeated. We observe that this *on-the-fly* data augmentation shows better performance than one-time batch data augmentation [3]. $\boldsymbol{c}_l$ is the attention context vector calculated by applying softmax to the attention weights [27]. $\boldsymbol{h}^{enc}[m]$ and $\boldsymbol{h}^{dec}_{l-1}$ are the encoder and the decoder hidden output vectors, respectively. $\boldsymbol{\beta}_l[m]$ is the attention weight feedback [27]. In [27], the peak value of the speech waveform is normalized to be one. However, since finding the peak sample value is not possible for on-line feature extraction, we did not perform this normalization. We modified the input pipeline so that the on-line feature generation can be performed. We disabled the clipping of feature range between -3 and 3, which is the default setting for Librispeech experiment in [27]. For better stability in LSTM training, we used the gradient clipping by global norm [29], which is implemented as `tf.clip_by_global_norm` API in Tensorflow [30]. We used six layers of bidirectional LSTM as our encoder and one layer of unidirectional LSTM layer in the decoder followed by a softmax layer.

4. Experimental Results

Table 1: *Word Error Rates (WERs) obtained using MFCC implemented in [31] and power mel filterbank coefficients on the Librispeech corpus [32]. For each WER number, the same experiment was conducted twice and averaged.*

Cell Size		MFCC	Power Mel Filterbank Coefficients
256 cell	test-clean	5.32 %	5.25 %
	test-other	16.82 %	16.40 %
	average	11.07 %	10.83 %
1024 cell	test-clean	4.19 %	4.21 %
	test-other	14.19 %	13.84 %
	average	9.19 %	9.03 %
1536 cell	test-clean	4.06 %	**3.94 %**
	test-other	13.97 %	**13.56 %**
	average	9.02 %	**8.75 %**

Table 2: *Word Error Rates (WERs) obtained with VTLP processing with different window lengths and different warping factor distribution range.*

Window Length		0.7 ∼ 1.3	0.8 ∼ 1.2	0.9 ∼ 1.1
25 ms	test-clean	4.03 %	3.80 %	3.80 %
	test-other	12.61 %	12.54 %	12.63 %
	average	8.32 %	8.17 %	8.22 %
50 ms	test-clean	3.82 %	**3.66 %**	3.86 %
	test-other	12.50 %	12.39 %	**12.35 %**
	average	8.16 %	**8.03 %**	8.11 %

Table 3: *Word Error Rates (WERs) obtained with VTLP processing with different window lengths and different warping factor distribution range with shallow-fusion using an RNN LM.*

Window Length		$0.7 \sim 1.3$	$0.8 \sim 1.2$	$0.9 \sim 1.1$
25 ms	test-clean	3.26 %	2.90 %	3.04 %
	test-other	10.60 %	10.40 %	10.40 %
	average	6.93 %	6.65 %	6.72 %
50 ms	test-clean	2.93 %	**2.85 %**	2.96 %
	test-other	10.40 %	10.25 %	**10.13 %**
	average	6.67 %	**6.55 %**	**6.55 %**

Table 4: *Word Error Rates (WERs) obtained with VTLP processing using shallow-fusion with a Transformer LM with different beam sizes. The window length is 50 ms, and the warping factor distribution is* $0.8 \sim 1.2$.

Beam Size	12	24	36	48
λ_p	0.005	0.004	0.003	0.002
λ_{lm}	0.45	0.46	0.48	0.48
test-clean	2.49 %	2.45 %	**2.44 %**	2.45 %
test-other	8.76 %	8.40 %	8.29 %	**8.22 %**
average	5.63 %	5.43 %	5.37 %	**5.34 %**

For speech recognition experiments, we used the Librispeech corpus [32] for training and evaluation. For training, we used the entire 960 hours training set consisting of 281,241 utterances. For evaluation, we used the official 5.4 hours test-clean and 5.1 hours test-other databases. In Table 1, we compared the performance between the baseline MFCC and the power-law of $(\cdot)^{\frac{1}{15}}$ features with different LSTM cell sizes. Especially for test-other, which is a more difficult task, the power mel filterbank coefficients shows better performance than the baseline MFCC. Thus, in all the following results in this section from Table 2 to Table 4, we use the power mel filter bank coefficients and use the LSTM cell size of 1536.

In Table 2, we show Word Error Rates (WERs) using different window sizes and warping coefficient distributions. In obtaining WERs in this table, we did not use any Language Models (LMs). The best performance was achieved when the window length is 50 *ms* and the warping coefficients are uniformly distributed between 0.8 and 1.2. We obtained 3.66 % WER on the *test-clean* database and 12.39 % WER on the *test-other* database. The results in Table 3 were obtained in the same configuration as that in Table 2, but we used shallow-fusion with an RNN LM. Using this shallow-fusion technique with an RNN-LM, we achieved 2.85 % WER and 10.25 % WER on the Librispeech *test-clean* and *test-other* sets.

Using the shallow-fusion with a Transformer LM [33, 34], we obtained significantly better result compared to those in Table 3 as shown in Table 4. In performing the shallow LM fusion, we used the following equation:

$$y^*_{0:L} = \arg\max_{y_{0:L}} \sum_{l=0}^{L-1} \Big[\log P\left(y_l | \boldsymbol{x}[0:M], y_{0:l}\right) - \lambda_p \log P(y_l) + \lambda_{\text{lm}} \log P\left(y_l | y_{0:l}\right) \Big], \quad (9)$$

which is slightly different from [35], where we have an additional term $\lambda_p \log P(y_l)$ for subtracting the prior bias that the model has learned from the training corpus. L is the length of the output label hypothesis. λ_p and λ_{lm} are weights for the prior probability and the LM prediction probability, respectively. In (9), we represented sequences following the Python slice notation. For example, $\boldsymbol{x}[0:M]$ denotes the sequence of the input acoustic features of length M, and $y_{0:L}$ is a sequence of output labels of length L. When the beam size is 36, $\lambda_p = 0.003$, and $\lambda_{\text{lm}} = 0.48$, we obtained 2.44 % WER on the *test-clean* set and 8.29 % WER on the *test-other* set.

5. Conclusions

In this paper, we presented an improved Vocal Tract Length Perturbation (VTLP) algorithm as a data augmentation technique. We perform frequency warping using a oversized FFT and a bilinear warping function. Time-domain waveform is resynthesized using OverLap Addition (OLA). This approach has two advantages compared to conventional approach of adjusting the center frequencies of mel filterbank during feature extraction. First, additional data augmentation using "acoustic simulator" [2, 3] is not affected by this frequency warping. Second, we may use a different window length for frequency warping from that used in feature processing. We observe that the best performance was obtained when the warping coefficient is uniformly distributed between 0.8 and 1.2, and the window length is 50 ms. We obtained 3.66 % WER and 12.39 % WER on the Librispeech *test-clean* and *test-other* using an attention-based end-to-end speech recognition system without using any Language Models (LMs). Using the shallow-fusion technique with a Transformer LM, we achieved 2.44 % WER and 8.29 % WER on the Librispeech *test-clean* and *test-other* databases. To the best of our knowledge, the 2.44 % WER on the *test-clean* is the best result ever reported on this database.

6. Acknowledgements

This research was funded by Samsung Electronics. The authors are thankful to Executive Vice President Seunghwan Cho, Shatrughan Singh, and all of the Speech Processing Lab. (SPL) members at Samsung Research.

7. References

[1] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Int. Conf. Mach. Learn. (ICML) Workshop on Deep Learn. Audio, Speech, Lang. Process.*, 2013.

[2] C. Kim, A. Misra, K.K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home," in *INTERSPEECH-2017*, Aug. 2017, pp. 379–383.

[3] C. Kim, E. Variani, A. Narayanan, and M. Bacchiani, "Efficient implementation of the room simulator for training deep neural network acoustic models," in *INTERSPEECH-2018*, Sept 2018, pp. 3028–3032. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-2566

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, Nov. 2012.

[5] T. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural

networks for automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, Feb. 2017.

[6] ——, "Raw Multichannel Processing Using Deep Neural Networks," in *New Era for Robust Speech Recognition: Exploiting Deep Learning*, S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, Ed. Springer, Oct. 2017.

[7] D. Gowda, A. Garg, K. Kim, M. Kumar, and C. Kim, "Multi-task multi-resolution char-to-bpe cross-attention decoder for end-to-end speech recognition," in *INTERSPEECH-2019*, Graz, Austria, Sept. 2019.

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

[9] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," in *INTERSPEECH-2017*, 2017, pp. 3707–3711. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1566

[10] C. Kim and R. M. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, pp. 1315–1329, July 2016.

[11] U. H. Yapanel and J. H. L. Hansen, "A new perceptually motivated MVDR-based acoustic front-end (PMVDR) for robust automatic speech recognition," *Speech Communication*, vol. 50, no. 2, pp. 142–152, Feb. 2008.

[12] T. Nakatani, N. Ito, T. Higuchi, S. Araki, and K. Kinoshita, "Integrating DNN-based and spatial clustering-based mask estimation for robust MVDR beamforming," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2017, pp. 286–290.

[13] C. Kim, K. Eom, J. Lee, and R. M. Stern, "Automatic selection of thresholds for signal separation algorithms based on interaural delay," in *INTERSPEECH-2010*, Sept. 2010, pp. 729–732.

[14] C. Kim, C. Khawand, and R. M. Stern, "Two-microphone source separation algorithm based on statistical modeling of angle distributions," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2012, pp. 4629–4632.

[15] C. Kim and R. M. Stern, "Nonlinear enhancement of onset for robust speech recognition," in *INTERSPEECH-2010*, Sept. 2010, pp. 2058–2061.

[16] C. Kim, K. Chin, M. Bacchiani, and R. M. Stern, "Robust speech recognition using temporal masking and thresholding algorithm," in *INTERSPEECH-2014*, Sept. 2014, pp. 2734–2738.

[17] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, Apr 1987, pp. 705–708.

[18] W. Hartmann, T. Ng, R. Hsiao, S. Tsakalidis, and R. Schwartz, "Two-stage data augmentation for low-resourced speech recognition," in *INTERSPEECH-2016*, 2016, pp. 2378–2382. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1386

[19] P. Zhan and A. Waibel, "Vocal tract length normalization for large vocabulary continuous speech recognition," School of Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-97-148, May 1997. [Online]. Available: https://www.lti.cs.cmu.edu/sites/default/files/CMU-LTI-97-150-T.pdf

[20] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sept 2015.

[21] C. Kim and R. M. Stern, "Power function-based power distribution normalization algorithm for robust speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 188–193.

[22] C. Kim, K. Kumar and R. M. Stern, "Robust speech recognition using small power boosting algorithm," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 243–248.

[23] C. Kim, K. Kumar, B. Raj, and R. M. Stern, "Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain," in *INTERSPEECH-2009*, Sept. 2009, pp. 2495–2498.

[24] C. Kim and R. M. Stern, "Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction," in *INTERSPEECH-2009*, Sept. 2009, pp. 28–31.

[25] A. V. Oppenheim and R. W.Scafer, with .J. R. Buck, *Discrete-time Signal Processing*, 2nd ed. Englewood-Cliffs, NJ: Prentice-Hall, 1998.

[26] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: the RWTH extensible training framework for universal recurrent neural networks," in *IEEE Int. Conf. Acoust., Speech, and Signal Processing*, March 2017, pp. 5345–5349.

[27] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *INTERSPEECH-2018*, 2018, pp. 7–11. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1616

[28] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (pncc) for robust speech recognition," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2012, pp. 4101–4104.

[29] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13. JMLR.org, 2013, pp. III–1310–III–1318. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043083

[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[31] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, K. Huff and J. Bergstra, Eds., 2015, pp. 18 – 25.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *IEEE Int. Conf. Acoust., Speech, and Signal Processing*, April 2015, pp. 5206–5210.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[34] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation," *CoRR*, vol. abs/1905.03072, 2019. [Online]. Available: http://arxiv.org/abs/1905.03072

[35] S. Toshniwal, A. Kannan, C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 369–375.