SPECTRAL DISTORTION MODEL FOR TRAINING PHASE-SENSITIVE DEEP-NEURAL NETWORKS FOR FAR-FIELD SPEECH RECOGNITION

Chanwoo Kim[1]*, Tara Sainath*[1]*, Arun Narayanan*[1] *Ananya Misra*[1]*, Rajeev Nongpiur*[2]*, and Michiel Bacchiani*[1]

[1]Google Speech, [2]Nest

{chanwcom, tsainath, arunnt, amisra, rnongpiur, michiel}@google.com

ABSTRACT

In this paper, we present an algorithm which introduces phase-perturbation to the training database when training phase-sensitive deep neural-network models. Traditional features such as log-mel or cepstral features do not have have any phase-relevant information. However features such as raw-waveform or complex spectra features contain phase-relevant information. Phase-sensitive features have the advantage of being able to detect differences in time of arrival across different microphone channels or frequency bands. However, compared to magnitude-based features, phase information is more sensitive to various kinds of distortions such as variations in microphone characteristics, reverberation, and so on. For traditional magnitude-based features, it is widely known that adding noise or reverberation, often called Multistyle-TRaining (MTR), improves robustness. In a similar spirit, we propose an algorithm which introduces spectral distortion to make the deep-learning models more robust to phase-distortion. We call this approach Spectral-Distortion TRaining (SDTR). In our experiments using a training set consisting of 22-million utterances with and without MTR, this approach reduces Word Error Rates (WERs) relatively by 3.2 % and 8.48 % respectively on test sets recorded on Google Home.

Index Terms— Far-field Speech Recognition, Deep-Neural Network Model, Phase-Sensitive Model Spectral Distortion Model, Spectral Distortion Training, Phase Distortion Training

1. INTRODUCTION

After the breakthrough of deep learning technology [1, 2, 3, 4, 5, 6], speech recognition accuracy has improved dramatically. Recently, speech recognition systems have begun to be employed not only in smart phones and Personal Computers (PCs) but also in standalone devices in far-field environments. Examples include voice assistant systems such as Amazon Alexa and Google Home [7, 8]. In 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 [9, 10], on-set enhancement algorithms [11, 12], and multi-microphone approaches [13, 14, 15, 16, 17].

It has been known that the Inter-microphone Time Delay (ITD) or Phase Difference (PD) between two microphones may be used to identify the Angle of Arrival (AoA) [18, 19]. The Inter-microphone Intensity Difference (IID) may also serve as a cue for determining the AoA [20, 21]. A different approach to this problem is using multi-channel features which contain temporal information between two microphones such as Complex Fast Fourier Transform (CFFT) [8, 7]. To train an acoustic model using these features, we need to collect a large number of utterances collected using that specific model of devices in real environments. Since multi-channel utterances have device-dependent characteristics such as the number of microphones and the distance between microphones, we need to re-collect multi-channel utterances for each device model. Thus, data collection is a critical problem for multi-channel features. To tackle this problem, we developed the "room simulator" [7] to generate simulated multi-microphone utterances for training multi-channel deep-neural network model. Multi-style Training (MTR) [22] driven by this room simulator was employed in training the acoustic model for Google Home [7, 8].

However, the room simulator in [7] still has its limitations. It assumes that all the microphones are ideal, which means that they all have zero-phase all-pass responses. Even though this assumption is very convenient, it is not true with actual microphones due to microphone spectrum distortion. In addition, there may be reasons for distortion such as electrical noise in the circuit, acoustic auralization effect from the hardware surface, and various vibrations. In conventional MTR, we usually only add additive noise and reverberation to the training set; we do not model the magnitude or phase distortion across different filter bank or microphone channels. In this paper, we propose an algorithm that makes phase-sensitive deep learning model more robust by adding phase distortion to the training set.

2. SPECTRAL-DISTORTION TRAINING (SDTR) FOR PHASE-SENSITIVE DEEP NEURAL NETWORKS

In this section, we explain the entire structure of Spectral-Distortion TRaining (SDTR), and its subsets Phase-Distortion TRaining (PDTR) and Magnitude Distortion TRaining (MDTR). PDTR is a subset of SDTR where distortion is only applied to the phase component without modifying the magnitude component of complex features. MDTR is a subset of SDTR where distortion is applied only to the magnitude component of such features. PDTR is devised for enhancing the robustness of phase-sensitive multi-microphone neural network models such as those presented in [8, 23].

2.1. Acoustic modeling with Spectral-Distortion TRaining (SDTR)

Fig. 1 shows the structure of the acoustic model pipeline using the SDTR to train multi-channel deep neural networks. The pipeline is based on our work described in [7, 8]. The first stage of the pipeline in Fig 1 is the room simulator to generate acoustically simulated utterances in millions of different virtual rooms [7]. To make the phase-sensitive multi-channel feature more robust, we add the Spectral Distortion Model (SDM) to each channel. Mathematically, SDM is described in (1). As input, we use the Complex Fast Fourier Transform (CFFT) feature whose window size is 32 *ms*, and the interval between successive frame is 10 *ms*. We use the FFT size of $N = 512$. Since FFT of real signals have Hermitian symmetry, we

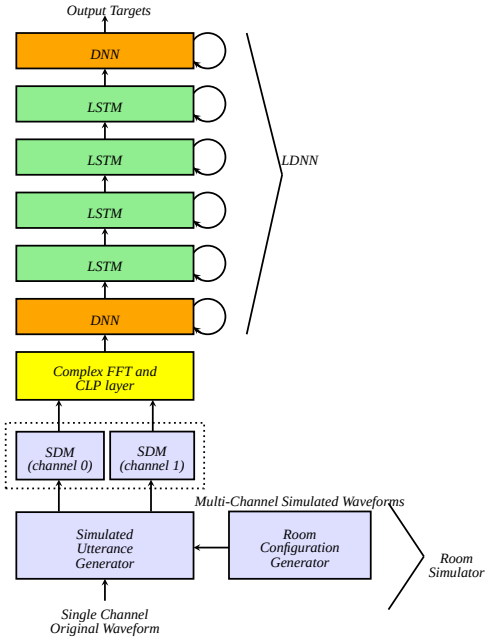

Fig. 1: A pipeline containing the Spectrum Distortion Model (SDM) (contained in the dashed box) for training deep-neural networks for acoustic modeling.

use the lower half spectrum whose size given by $N/2 + 1 = 257$. Since it has been shown that long-duration features represented by overlapping features are helpful [24], four frames are stacked together and the input is downsampled by a factor of 3. Thus we use a context dependent feature consisting of 2056 complex numbers given by 257 (the size of the lower half spectrum) x 2 (number of channels) x 4 (number of stacked frames). The acoustic model is the factored complex linear projection (fCLP) model described in [8]. fCLP model passes the CFFT features to complex valued linear layers that mimic filter-and-sum operation in the spectral domain. The output is then passed to a complex linear projection layer [25], followed by a typical multi-layer Long Short-Term Memory (LSTM) [26, 27] acoustic model. We use 4-layer LSTM with 1024 units in each layer. The output of the final LSTM layer is passed to a 1024 unit Deep Neural Network (DNN), followed by a softmax layer. The softmax layer has 8192 nodes corresponding to the number of tied context-dependent phones in our ASR system. The output state label is delayed by five frames, since it was observed that the information about future frames improves the prediction of the current frame [7, 8].

2.2. Spectral Distortion Model (SDM)

The spectrum distortion procedure is summarized by the following pseudo-code:

for each utterance in the training set **do**
  for each microphone channel of the utterance **do**
    Create a random Spectral Distortion Model (SDM) using (1).
    Perform Short-Time Fourier Transform (STFT).
    Apply this transfer function to the spectrum.
    Re-synthesize the output microphone-channel using Over-Lap Addition (OLA).
  end for
end for

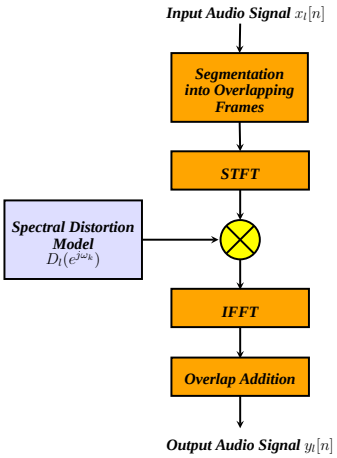

Fig. 2: A diagram showing the structure of applying Spectrum Distortion Model (SDM) in (1) to each microphone channel. Note that l in this diagram denotes the microphone channel index.

For each microphone channel of each utterance, we create a single Spectral Distortion Model (SDM). This random model is not regenerated for each frame. The Spectral Distortion Model (SDM) is described by the following equation:

$$D_l(e^{j\omega_k}) = e^{am_l(k)+jp_l(k)}, \qquad 0 \le k \le \frac{K}{2}, \quad 0 \le l \le L-1. \tag{1}$$

where l is the microphone channel index and L is the number of microphone channels. In the case of Google Home, since we use two microphones, $L = 2$. k is the discrete frequency index, ω_k is defined by $\omega_k = \frac{2\pi k}{K}$ where K is the Discrete Fourier Transform(DFT) size. $m_l(k)$ and $p_l(k)$ are Gaussian random samples pulled from the following Gaussian distributions $\mathbf{m}$ and $\mathbf{p}$ respectively:

$$\mathbf{m} \sim \mathcal{N}(0, \sigma_m^2) \tag{2a}$$

$$\mathbf{p} \sim \mathcal{N}(0, \sigma_p^2) \tag{2b}$$

The scaling coefficient a in (1) is defined by the following equation:

$$a = \ln(10.0)/20.0 \tag{3}$$

This scaling coefficient a is introduced to make σ_m the standard deviation of the magnitude in decibels, which makes it easier to control the amount of distortion. From (1), it should be evident that $m_l(k)$ and $p_l(k)$ are related to the magnitude and phase distortion, respectively. The magnitude distortion is accomplished by the $e^{am_l(k)}$ term. Using the properties of logarithm, we observe that the standard deviation of magnitude in decibel $\left(20\log_{10}\left|D_l(e^{j\omega_k})\right|\right)$ is given by σ_m. For the phase term, since the complex exponential has a period of 2π, the distribution actually becomes the wrapped Gaussian distribution [28].

After creating the spectrum distortion transfer function $D_l\left(e^{j\omega_k}\right)$ in (1), we process each channel using the structure shown in Fig. 2. We apply the Hanning window instead of the more frequently-used Hamming window to each frame. We use the Hanning window

to better satisfy the OverLap-Add (OLA) constraint. After multiplying the complex spectrum of each frame with the spectrum distortion transfer function $D_l\left(e^{j\omega_k}\right)$ in the frequency domain, the time-domain signal is re-synthesized using OverLap-Add (OLA) synthesis. This processing is shown in detail in Fig. 2. The reason for going back to the time domain is because we use Complex Fast Fourier Transform (CFFT) as feature whose frame size is 32 *ms* in Fig. 1, which does not match the processing window size of SDM. We segment each microphone channels into successive frames with the frame length of 10 *ms*. The period between successive frames is 5 *ms*. These frame length is chosen based on the experimental results in Sec. 2.3. The spectrum distortion effects from $D_l(e^{j\omega_k})$ in Fig. 2 is not removed by either the conventional Causal Mean Subtraction (CMS) [29], nor Cepstral Mean Normalization (CMN). This is because our feature and the SDM model are complex numbers and functions, and CMS and/or CMN operates on the magnitude.

2.3. Word Error Rate(WER) dependence on σ_m, σ_p and frame length

Table 1 shows speech recognition results in terms of Word Error Rate (WER) using PDTR with different values of σ_p and frame lengths. The configurations for speech recognition training and evaluation will be described in detail in Sec. 3. The evaluation set used in Table 1 through Table 4 is the combinations of five rerecording sets described in Sec. 3, which are three rerecording sets using different Google Home devices, and two rerecording sets in presence of Youtube noise and interfering speakers. The best result in Table 1 (49.77 % WER) is obtained when $\sigma_p = \infty$ with the window length of 32 *ms*. Table 2 shows Word Error Rates (WERs) using MDTR on the same test set using the same configuration as in Table 1 with different σ_m values. In these experiments, we observe significant improvement with PDTR and MDTR over the baseline system, which shows WER of 62.0 % on the same test set.

When training acoustic models for Google Home, we have been using data generated by the room simulator [7]. Table 3 and Table 4 show the WERs when the PDTR or MDTR is applied with the Multi-style TRaining (MTR) driven by this "room simulator". Even though relative improvement over the MTR baseline in Table 3 and Table 4 is less than the relative improvement in Table 1 and Table 2, we still obtain substantial improvement over the baseline.

From the results from Table 1 to Table 4, we observe that PDTR is more effective than MDTR in our acoustic model using CFFT feature. We also tried combinations of PDTR and MDTR, but we could not obtain results better than only using PDTR. Thus, in the final system, we adopt PDTR with $\sigma_p = 0.4$ as the default Spectral Distortion Model (SDM) in (1).

Table 1: Word Error Rates (WERs) using the PDTR training

	baseline	$\sigma_p = 0.1$	$\sigma_p = 0.4$	$\sigma_p = \infty$
frame length				
10 *ms*	62.00%	57.16 %	56.74 %	54.03 %
32 *ms*		59.03 %	57.14 %	**49.77** %

Table 2: Word Error Rates (WERs) using the MDTR training

	baseline	$\sigma_m = 0.5$	$\sigma_m = 1.0$	$\sigma_m = 2.0$
frame length				
10 *ms*	62.00%	60.39 %		
32 *ms*		**52.21** %	53.03 %	55.37 %

Table 3: Word Error Rates (WERs) using the PDTR and MTR training

	MTR baseline	$\sigma_p = 0.1$	$\sigma_p = 0.4$	$\sigma_p = \infty$
frame length				
10 *ms*		28.63 %	**28.40** %	29.78 %
32 *ms*	29.34%		29.28 %	30.34 %
160 *ms*		28.69 %	31.36 %	37.82 %

Table 4: Word Error Rates (WERs) using the MDTR and MTR training

	MTR baseline	$\sigma_m = 0.5$	$\sigma_m = 1.0$	$\sigma_m = 2.0$
frame length				
10 *ms*		31.13 %		
32 *ms*	29.34%	**28.46** %	28.78 %	28.70 %
160 *ms*			29.01 %	29.55 %

3. EXPERIMENTAL RESULTS

In this section, we shows experimental results obtained with the SDTR training. For training, we used an anonymized 22-million English utterances (18,000-hr), which are hand-transcribed. For training the acoustic model, instead of directly using these utterances, we use the room simulator described in [7] to generate acoustically simulated utterances for our hardware. In the simulator, we use the 7.1 cm distance between two microphones. For each utterance, one room configuration was selected out of three million room configurations with varying room dimension, and varying the target speaker and noise source locations. In each room, number of noise sources may be up to three. This configuration changes for each training utterance. After every epoch, we apply a different room configuration to the utterance so that each utterance may be regenerated in somewhat different ways. For additive noise, we used Youtube videos, recordings of daily activities, and recordings at various locations inside cafes. We picked up the SNR value from a distribution ranging from 0 *dB* to 30 *dB*, with an average of 11.08 *dB*. We used reverberation time varying from 0 *ms* up to 900.0 ms with an average of 482 *ms*. To model reverberation, we employed the image method [30]. We constructed $17^3 - 1 = 4912$ virtual sources for each real sound source. The acoustic model was trained using the Cross-Entropy (CE) minimization as the objective function after aligning each utterance. The Word Error Rates (WERs) are obtained after 120 million steps of acoustic model training.

For evaluation, we used around 15-hour of utterances (13,795 utterances) obtained from anonymized voice search data. Since our objective is evaluating speech recognition performance when our system is deployed on the actual hardware, we re-recorded these utterances using our actual devices in a real room at five different locations. The utterances were played out using a mouth simulator. We used three different devices (named "Device 1", "Device 2", and "Device 3") as shown in Table 5 and 6. These three devices

Table 5: Word Error Rates (WERs) obtained with the PDTR ($\sigma_m = 0.0,\ \sigma_p = 0.4$) training

	Baseline	PDTR	Relative improvement (%)
Original Test Set	12.02 %	12.32 %	-2.53 %
Simulated Noise Set 1	20.34 %	20.72 %	-1.86 %
Simulated Noise Set 2	47.88 %	46.69 %	2.50 %
Rerecording using "Device 1"	50.14 %	42.87 %	14.51 %
Rerecording using "Device 2"	48.65 %	43.32 %	10.95 %
Rerecording using "Device 3"	56.27 %	51.30 %	8.83 %
Rerecording with youtube background noise	76.01 %	71.42 %	6.04 %
Rerecording with multiple interfering speaker noise	78.95 %	74.80 %	5.26 %
Average from rerecording sets	**62.00** %	**56.74** %	**8.48** %

Table 6: Word Error Rates (WERs) obtained with the PDTR ($\sigma_m = 0.0,\ \sigma_p = 0.4$) training combined with room-simulator based MTR in [7]

	MTR	PDTR + MTR	Relative improvement (%)
Original Test Set	11.97 %	11.99 %	-0.17 %
Simulated Noise Set 1	14.73 %	15.03 %	-2.04 %
Simulated Noise Set 2	19.55 %	20.29 %	-3.79 %
Rerecording using "Device 1"	21.89 %	20.86 %	4.71 %
Rerecording using "Device 2"	22.23 %	21.29 %	4.22 %
Rerecording using "Device 3"	22.05 %	21.65 %	1.81 %
Rerecording with youtube background Noise	34.83 %	34.21 %	1.78 %
Rerecording with multiple interfering speaker noise	44.79 %	44.00 %	1.76 %
Average from rerecording sets	**29.34** %	**28.40** %	**3.20** %

are prototype Google Home devices. Each device is placed at five different positions and orientations in a real room with mild reverberation (around 200 *ms* reverberation time). The entire 15-hour test utterances are rerecorded using each device. We also prepared two additional rerecorded sets in presence of Youtube noise and interfering speaker noise played through real loud speakers. The noise level varies, but it is usually between 0 and 15 dB SNR. Each of these noisy rerecording sets also contains the same 15-hour long utterances with subsets being recorded at five different locations. In total, there are five rerecording test sets in Table 5 and Table 6. In addition to the real rerecorded sets, we evaluated performance on two simulated noise sets created using the same utterances using the "room simulator" in [7]. Note that in these two simulated noise sets, we assume that all microphones are identical without any magnitude or phase distortion. We are mainly interested in performance on the rerecorded sets, but we also included these simulated noise sets for the purpose of comparison.

In Table 5, we compare the performance of the baseline system with the PDTR system. The baseline Word Error Rates (WERs) are high on rerecorded test sets because the baseline system was not processed by MTR using the room simulator in [7]. Based on our analaysis in Sec. 2, we use the PDTR of $\sigma_m = 0.0, \sigma_p = 0.4$ in (2) as our Spectral Distortion Model (SDM). As shown in these two tables, PDTR shows significantly better results than the baseline for rerecorded sets while doing on par or slightly worse on two simulated noisy sets, which is expected.

As shown in Tables 5 and 6, the final system shows relatively 8.48 % WER reduction for the non-MTR training case and relatively 3.2 % WER reduction for the MTR training case using the room simulator described in [7].

4. CONCLUSIONS

In this paper, we described Spectral Distortion TRaining (SDTR) and its subsets Phase Distortion TRaining (PDTR) and Magnitude Distortion TRaining (MDTR). These training approaches apply the Spectral Distortion Model (SDM) to each microphone channel of each training utterance. This algorithm is developed to make the phase-sensitive neural net model robust against various distortions in signals. Our experimental results show that the phase-sensitive neural-net trained with PDTR is much more robust against real-world distortions. The final system shows relatively 3.2 % WER reduction over the MTR training set in [7] for Google Home.

5. REFERENCES

[1] M. Seltzer, D. Yu, and Y.-Q. Wang, "An investigation of deep

neural networks for noise robust speech recognition," in *Int. Conf. Acoust. Speech, and Signal Processing*, 2013, pp. 7398–7402.

[2] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks - studies on speech recognition tasks," in *Proceedings of the International Conference on Learning Representations*, 2013.

[3] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011.

[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.

[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] 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.

[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] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] T. Higuchi and N. Ito and T. Yoshioka and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2016, pp. 5210–5214.

[15] H. Erdogan, J. R. Hershey, S. Watanabe, M. Mandel, J. Roux, "Improved MVDR Beamforming Using Single-Channel Mask Prediction Networks," in *INTERSPEECH-2016*, Sept 2016, pp. 1981–1985.

[16] 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.

[17] R. M. Stern, E. Gouvea, C. Kim, K. Kumar, and H .Park, "Binaural and multiple-microphone signal processing motivated by auditory perception," in *Hands-Free Speech Communication and Microphone Arrays, 2008*, May. 2008, pp. 98–103.

[18] 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.

[19] C. Kim, K. Kumar, and R. M. Stern, "Binaural sound source separation motivated by auditory processing," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2011, pp. 5072–5075.

[20] H. S. Colburn and A. Kulkarni, "Models of sound localization," in *Sound Source Localization*, A. N. Popper and R. R. Fay, Eds. Springer-Verlag, 2005, pp. 272–316.

[21] N. Roman, D. Wang, and G. Brown, "Speech segregation based on sound localization," *The Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2236–2252, 2003.

[22] 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.

[23] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, and M. Bacchiani, "Learning the Speech Front-end With Raw Waveform CLDNNs," in *INTERSPEECH-2015*, Sept. 2015, pp. 1–5.

[24] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," in *INTERSPEECH-2015*, Sept. 2015, pp. 1468–1472.

[25] E. Variani, T. Sainath, I. Shafran, and M. Bacchiani, "Complex Linear Projection (CLP): A Discriminative Approach to Joint Feature Extraction and Acoustic Modeling," in *INTERSPEECH-2016*, Sept. 2016, pp. 808–812.

[26] S. Hochreiter and Jürgen Schmidhuber, "Long Short-term Memory," *Neural Computation*, no. 9, pp. 1735–1780, Nov. 1997.

[27] T. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 2015, pp. 4580–4584.

[28] E. Breitenberger, "Analogues of the normal distribution on the circle and the sphere," *Biometrika*, vol. 50, no. 1/2, pp. 81–88, June 1963.

[29] B. King, I. Chen, Y. Vaizman, Y. Liu, R. Maas, S. Parthasarathi, B. Hoffmeister, "Robust Speech Recognition via Anchor Word Representations," in *INTERSPEECH-2017*, Aug. 2017, pp. 2471–2475.

[30] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, April 1979.