Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home

Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes,
Arun Narayanan, Tara Sainath, and Michiel Bacchiani

Google Speech
{chanwcom, amisra, kkchin, thadh, arunnt, tsainath, michiel}@google.com

Abstract

We describe the structure and application of an acoustic room simulator to generate large-scale simulated data for training deep neural networks for far-field speech recognition. The system simulates millions of different room dimensions, a wide distribution of reverberation time and signal-to-noise ratios, and a range of microphone and sound source locations. We start with a relatively clean training set as the source and artificially create simulated data by randomly sampling a noise configuration for every new training example. As a result, the acoustic model is trained using examples that are virtually never repeated. We evaluate performance of this approach based on room simulation using a factored complex Fast Fourier Transform (CFFT) acoustic model introduced by Sainath et. al. (2016), which uses CFFT layers and LSTM AMs for joint multi-channel processing and acoustic modeling. Results show that the simulator-driven approach is quite effective in obtaining large improvements not only in simulated test conditions, but also in real / rerecorded conditions. This room simulation system has been employed in training acoustic models including the ones for the recently released Google Home.

Index Terms: Simulated data, room acoustics, robust speech recognition, deep learning

1. Introduction

Recent advances in deep-learning techniques and the availability of large training databases have resulted in significant improvements in speech recognition accuracy [1, 2]. Speech recognition systems that use deep-neural network (DNN) has shown much better performance than those that use the traditional Gaussian Mixture Model (GMM). Nevertheless, such systems still remain sensitive to mismatch in training and test conditions. The presence of additive noise, channel distortion, and reverberation can cause such mismatch.

Traditional approaches for enhancing speech recognition accuracy in the presence of noise include beam forming [3], masking [4, 5, 4, 6, 7], and robust feature extraction [8, 9, 10, 11]. But recent results have shown that robustness of DNN-based models largely depends on the quality of the data that it is trained on [12]. Typically, using a training set that matches the final test conditions results in largest improvements in performance. However, in many cases it may not be practical to obtain such a set. For example, Google recently released a new commercial product, Google Home, that targets far-field use cases. Before releasing the product, it was hard to obtain a large enough training set that closely matches this use case. In such cases, deriving a set from existing training sets via simulation is a reasonable compromise. The quality of the model trained on derived sets depends on how good the simulation is, and how closely it captures the wide variety of use cases. For such use cases, we developed a simulation system that synthesizes speech utterances in a wide variety of conditions – rooms with varying dimensions, noise levels, reverberation time, target speaker and noise locations, and number of noise sources. This room simulation system processes relatively clean, 1-channel utterance to create simulated noisy far-field utterances.

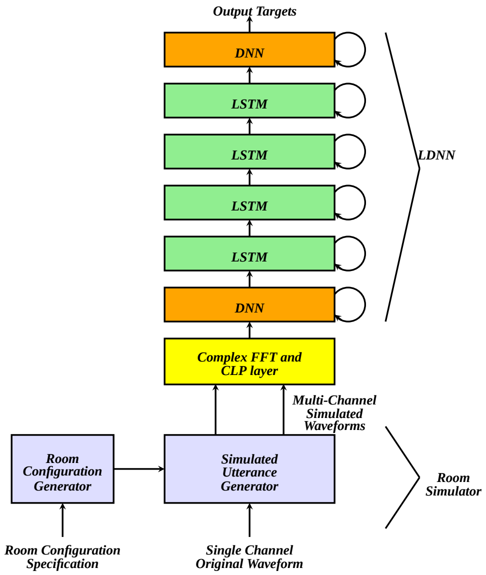

Output Targets
DNN
LSTM
LSTM
LSTM
LSTM
DNN
LDNN
Complex FFT and CLP layer
Multi-Channel Simulated Waveforms
Room Configuration Generator
Simulated Utterance Generator
Room Simulator
Room Configuration Specification
Single Channel Original Waveform

Figure 1: An LDNN training architecture [13] using simulated utterance data.

The rest of the paper is organized as follows. We describe the room simulator and acoustic model training using simulated data in the following section. Experimental results that demonstrate the utility of simulated data is presented in Section 3. We conclude in Section 4.

2. Training using simulated data

An overview of the entire training system is shown in Fig. 1. As mentioned, the goal of simulation is to create artificial utterances that mimic real use cases like far-field speech recognition. The simulation uses an existing corpus of relatively clean utterances as the source. Typically, the only prior knowledge we have for simulation are the hardware characteristics like microphone spacing, and targeted use-case scenarios like expected room size distributions, reverberation times, background noise levels, and target to microphone array distances. "Room configuration generator” in Fig. 1 generates a random configuration based on the available prior knowledge. The configuration and a pseudo-clean utterance from the source corpus are passed to a “Room simulator” that performs the necessary simulation. The output of the room simulator is a multi-channel simulated noisy utterance, which is then fed as input to an acoustic model. For the experiments in this paper, we use a factored CFFT LDNN acoustic model, introduced in [14]. The model performs multi-channel processing and acoustic modeling jointly, and has been shown to provide comparable or better performance to traditional enhancement techniques like beamforming [15].

We describe the individual components of the system in detail in the following subsections.

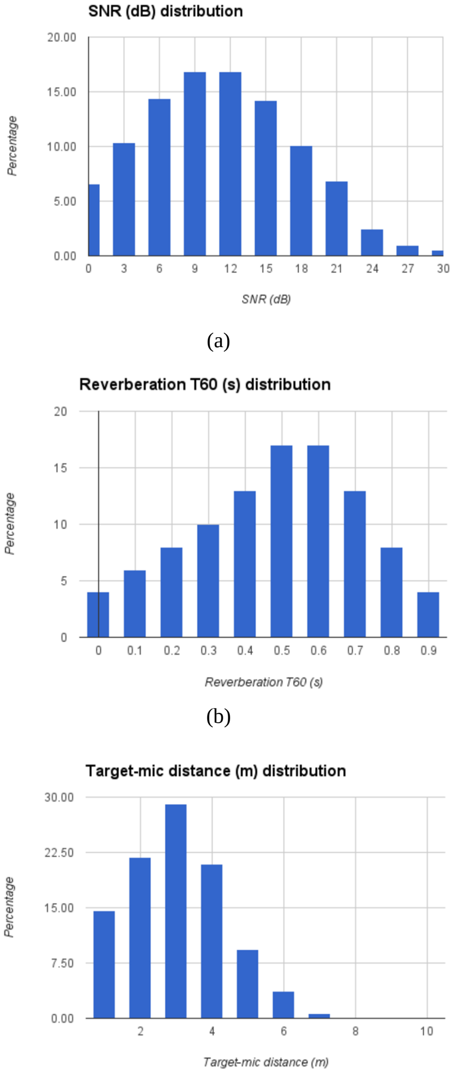
2.1. Room configuration generation

The room configuration generator takes distributions of prior knowledge as input and outputs an arbitrary number of randomly generated room configurations. Specifically for the Google Home use case, we created 3,000,100 room configurations to include a large number of room sizes, source positions, signal to noise ratios (SNRs), reverberation times, and number of noise sources. The size of the room was randomly set to have a width uniformly between 3 meters to 10 meters, and a length between 3 meters to 8 meters and a height between 2.5 meters to 6 meters. Within the room, the target and noise source locations are randomly selected with respect to the microphone. For the target source, the azimuth, θ, and elevation, ϕ, are randomly selected to be in the interval $[-180.0^o, 180.0^o]$ and $[45.0^o, 135.0^o]$, respectively. The number of noise sources is randomly set to be between zero and three. The location of the noise source is also randomly selected, but the distribution of θ and ϕ is constrained to be between -180.0^o and 180.0^o and -30.0^o and 180.0^o, respectively. We intentionally set the distribution of the noise sources to be wider than that of the target source. When the sound source locations (target or nosie) are chosen, we assume that they are at least 0.5 meters away from the wall. The noise source intensities are set so that the utterance level Signal-to-noise Ratio (SNR) is between 0 dB and 30 dB, with an average of 12 dB over the entire corpus. The SNR distribution was also conditioned on the target to mic distance. Fig. 2(a) shows the distribution of the SNR from which the SNR level of specific utterance is pulled. Reverberation of each room is randomly chosen to be between 0 milliseconds (no reverberation) and 900 milliseconds. Fig. 2(b) shows the distribution of the reverberation time given in T_{60}.

The distributions were chosen with two goals in mind: 1) The distribution should cover a wide range of use cases, and 2) The distribution should be biased towards the typical use cases the device is targeting. We did not spend a lot of effort trying to tune the parameters, but rather chose a wide range in order to make sure that the acoustic model sees a high level of variation in the simulated training data.

2.2. Room Simulation

Fig. 3(a) shows an illustration of a room structure in our room simulator. The room is assumed to be a cuboid with one or more microphones in the room. There is exactly one target sound source. There may be zero, one, or multiple noise sound sources inside the same room. All sound sources are assumed to be dirctional. In Fig. 3(a), the target sound source and noise sound sources are represented by a black ball and light gray balls, respectively.



(a)

(b)

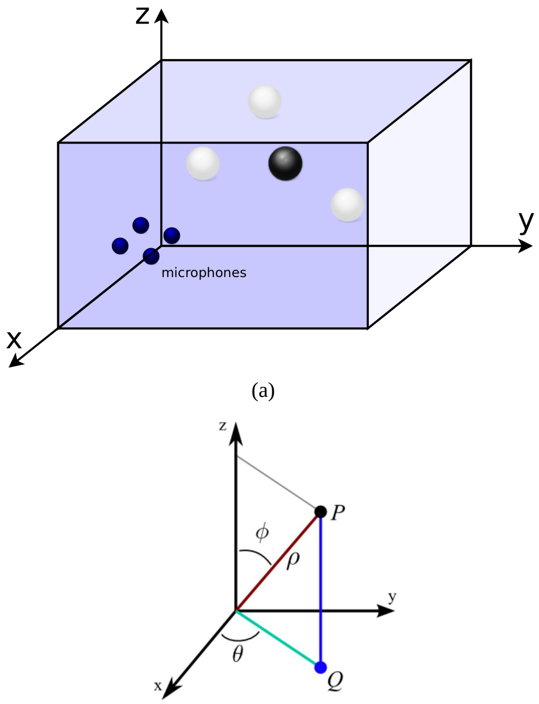
(c)

Figure 2: *(a) The SNR distribution, (b) The reverberation time (T_{60}) distribution, and (c) The distribution of the distance from the target source to microphone.*

Assuming that there are I sound sources including one target source, and J microphones, and assuming that acoustic reflection inside a room is modeled by a Linear Time-Invariant(LTI) system, the received signal at microphone $j, 0 \leq j < J$ is expressed by the following equation:

$$y_j[n] = \sum_{i=0}^{I-1} (\alpha_{ij} h_{ij}[n] * x_i[n]) \qquad (1)$$

where $x_i[n], 0 \leq i < I$ is a sound source, and α_{ij} are coefficients that control signal level. Among the sound sources, we define $x_0[n]$ to be the target sound source, while the remaining $x_i[n], \quad 1 \leq i < I$ are noise sound sources. To reduce the computational cost, all computations are performed in (1) in the frequency domain.



(a)

(b)

Figure 3: *(a) A simulated room: There may be multiple microphones, a single target sound source, multiple noise sources in a cuboid-shape room with acoustically reflective walls. (b) Spherical coordinates.*

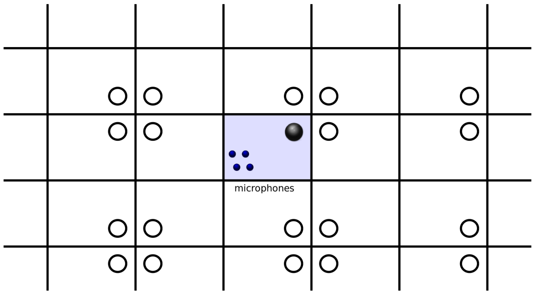


Figure 4: *A diagram showing the location of the real and the virtual sound sources assuming that the walls reflects acoustic wave like a mirror*

2.3. Room impulse response modeling

We use the image method to model the room impulse responses $h_{ij}[n]$ [16, 17, 18] in (1). Given the sound source position, the microphone position, and the the reverberation time, we obtain the location of impulses by calculating the distance between the microphone and the real and the virtual sound sources shown in Fig. 4. Following the image method, the impulse response is calculated using the following equation [16, 17]:

$$h[n] = \sum_{i=0}^{I-1} \frac{r^{g_i}}{d_i} \delta \left[n - \left\lceil \frac{d_i f_i}{c_0} \right\rceil \right] \qquad (2)$$

where i is the index of each virtual sound source, and d_i is the distance from that sound source to the microphone, r is the reflection coefficient of the wall , and g_i is the number of the reflections to that sound source, and c_0 is the speed of sound in the air. To reduce the computational cost of the room impulse response calculation in (2), we adopted the efficient RIR calculation approach proposed by S. McGovern [18]. Reverberation is controlled via the reflection coefficient r. To convert a randomly sampled T_{60} time into the reflection coefficient r, we use the Eyring's empirical equation in the inverse form [19]:

$$\alpha = 1 - \exp\left(-0.16\frac{V}{ST_{60}})\right), \qquad (3)$$

$$r = \sqrt{1 - \alpha^2}, \qquad (4)$$

where V and S are the volume and the total surface area of the room, respectively. I in (2) is the total number of true and virtual sound sources for a single true sound source. This can be set arbitrarily, but the chosen value will directly affect computation speed. In training the acoustic model for Google Home, we used $I = 17^3 = 4912$ sound sources including one true source for each true sound source.

Even though speech samples are usually sampled at either 8 kHz or 16 kHz, we need much higher sampling rate for the RIR creation in (2). This is because $h[n]$ in (2) is a series of impulses, and time delay difference the simulator can model is limited by the time delay between two adjacent impulses. If we represent the time delay between two impulses as τ, then the relationship between the angle θ and τ is given by the following equation:

$$\theta = \arcsin\left(\frac{c_{air}\tau}{f_s d}\right). \qquad (5)$$

Form the above equation, to have at least θ_0 resolution for 1-sample delay, the sampling rate should satisfy the following equation:

$$f_s \geq \frac{c_{air}}{d\sin(\theta_0)} \qquad (6)$$

To have 0.5-degree resolution assuming the microphone spacing of 7.1-cm, the above equation yields 548.8 kHz. To have some margin for other microphone spacing, in our room simulator, the default impulse response is sampled at $1,024$-kHz.

One of the limitations of the image method is if there is strong reverberation, then it introduces an unwanted low-frequency component as shown in Fig. 6(c) [16]. In speech recognition experiments, we found that this low frequency component does no harm since the frequency of such components are usually lower than 10.0 Hz. But to remove this low frequency components for other cases, we optionally process a linear-phase Finite-duration Impulse Response (FIR) filter after down-sampling to 128 kHz. We chose the cut-off frequency of 80-Hz.

After the aforementioned pre-processing, the final RIR is down-sampled to have the same sampling rate as the speech signal. For the Google Home use case, we assume two microphones with a mic-spacing of 71 millimeters.

2.4. Acoustic model training

For the experiments, the simulated data is used to train a CFFT factored complex linear projection (fCLP) LDNN acoustic model. fCLP LDNN is chosen as it takes complex FFT as input with the potential of doing implicity multichannel spatial processing. the fCLP layer uses 5 look directions and a 128 dimensional CLP layer. This is followed by a low rank projection

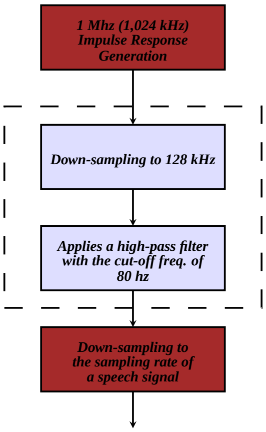
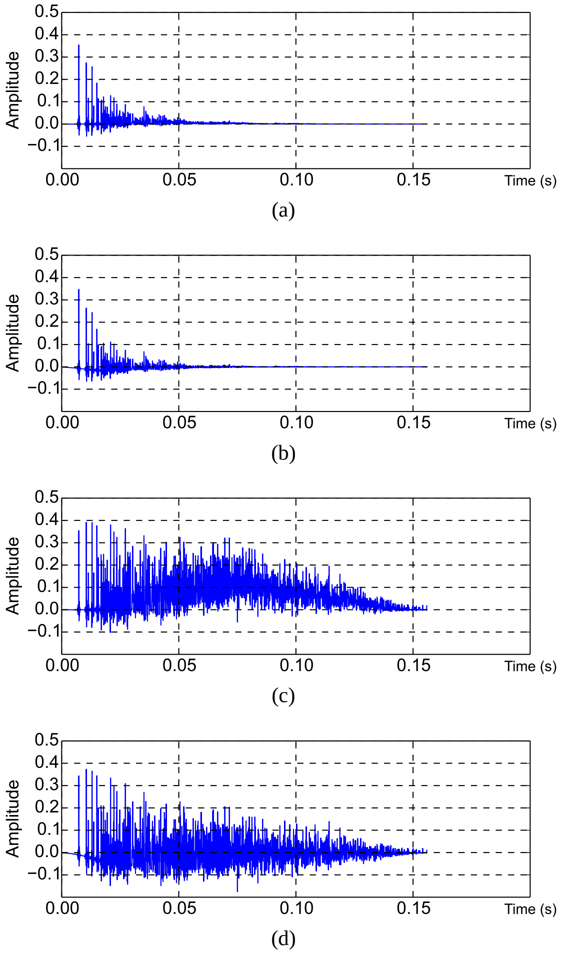

1 Mhz (1,024 kHz) Impulse Response Generation
Down-sampling to 128 kHz
Applies a high-pass filter with the cut-off freq. of 80 hz
Down-sampling to the sampling rate of a speech signal

Figure 5: The procedure for calculating Room Impulse Responses (RIRs).



(a)

(b)

(c)

(d)

Figure 6: *(a) Simulated room impulse responses. (a) Small reverberation (T_{60} = 70 ms) without high-pass filtering. (b) Small reverberation (T_{60} = 70 ms) with a sharp linear-phase high -pass filtering. (c) Strong reverberation (T_{60} = 400 ms) without high-pass filtering. (d) Strong reverberation (T_{60} = 400 ms) with a sharp linear-phase high-pass filtering.*

and the LDNN acoustic model. We use four layers of LSTMs each of which has 1024 units, and a 1024 dimensional DNN before the final softmax layer (see [20] for additional details).

Table 1: *Speech recognition experimental result.*

	Trained with the room simulator Word Error Rates	Baseline system Word Error Rates
Original Test Set	13.18 %	13.27 %
Simulated Noise Set	22.19 %	47.02 %
Device 1	23.98 %	36.09 %
Device 2	24.35 %	37.02 %
Device 3	24.18 %	36.32 %
Device 3 (Noisy Condition)	37.28 %	58.24 %
Device 3 (Mult-italker Condition)	47.56 %	61.92 %

3. Experimental results

In this section, we show speech recognition experimental results with and without using the room simulator. For the recognition system using the room simulator, we used the configuration described in Sec. 2.1. For the baseline system, we used the same pipeline in Fig. 1, but we exclude the room simulation stage – in effect, the model is trained on pseudo-clean utterances. For this baseline system, we replicate the original signal-channel audio to make two-channel input.

For training the acoustic model, we used anonymized 15,000-hour English utterances (22-million utterances), which are hand-transcribed. The acoustic model is trained to minimize the Cross-Entropy (CE) as the objective function. The supervised labels for CE training are always generated from the clean utterance. We chose not to use more complex acoustic models and skipped the sequence training step for faster turnaround times. We expect the gains to remain even after these stages as they are somewhat orthogonal to the quality of training data. The main goal of the experimental results in this section is to compare Word Error Rates (WERs) between system trained w/ and w/o simulated far-field data.

To evaluate our speech recognizer, we used an evaluation set of around 15-hour of utterances (13,795 utterances) obtained from anonymized voice search queries. Since our objective is deploying our speech recognition systems on far-field standalone devices such as Google Home, we rerecorded this evaluation set using the actual hardware in far-field environment. Note that the actual Google Home hardware has two microphones with microphone spacing of $7.1cm$, which matches the configuration of the room simulator. Three different devices were used in rerecording, and each device was placed in five different locations in an actual room resembling a real living room. 1. As shown in 1, the acoustic model trained using simulated utterances performs much better than the original baseline.

4. Conclusions

In this paper, we described our system to simulate millions of different utterances in millions of virtual rooms. We described the design decisions of the simulator and showed how we used this system to train deep-neural network models. This simulation based approach was employed in our Google Home product and brought significant performance improvement.

5. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. 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.

[2] 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.

[3] 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.

[4] C. Kim and K. K. Chin, "Sound source separation algorithm using phase difference and angle distribution modeling near the target," in *INTERSPEECH-2015*, Sept. 2015, pp. 751–755.

[5] C. Kim, K. 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.

[6] 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.

[7] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Int. Conf. Acoust. Speech, and Signal Processing*, 2013, pp. 7092–7096.

[8] 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.

[9] 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.

[10] ——, "Feature extraction for robust speech recognition based on maximizing the sharpness of the power distribution and on power flooring," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2010, pp. 4574–4577.

[11] ——, "Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction," in *INTERSPEECH-2009*, Sept. 2009, pp. 28–31.

[12] 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.

[13] 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.

[14] "T. Sainath, R. Weiss, K. Wilson, A. Narayanan, and M. Bacchiani", ""Factored spatial and spectral multichannel raw waveform CLDNNs"," in *"IEEE Int. Conf. Acoust., Speech, Signal Processing"*, March 2016, pp. 5075–5079.

[15] T. N. 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.

[16] 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.

[17] E. A. Lehmann, A. M. Johansson, and S. Nordholm, "Reverberation-time prediction method for room impulse responses simulated with the image-source model," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2007, pp. 159–162.

[18] S. G. McGovern. A model for room acoustics. [Online]. Available: http://www.sgm-audio.com/research/rir/rir.html

[19] L. Beranek, "Analysis of Sabine and Eyring equations and their application to concert hall audience and chair absorption." *"The Journal of the Acoustical Society of America"*, pp. 1399–1410, June.

[20] B. Li, T. Sainath, J. Caroselli, A.Narayanan, M. Bacchiani, A. Misra, I. Shafran, G. Pundak, K.K. Chin, K-C Sim, R. Weiss, K. Wilson, E. Variani, /chanwcom, O. Siohan, M. Weintraub, E. McDermott and R. Rose, ""Acoustic modeling for Google Home"," in *INTERSPEECH-2017*, Aug. 2017, p. (submitted).