Multi-Microphone Adaptive Noise Cancellation for Robust Hotword Detection

Yiteng (Arden) Huang, Turaj Z. Shabestary, Alexander Gruenstein, Li Wan

Google Inc., USA

{ardenhuang, turajs, alexgru, liwan}@google.com

Abstract

Recently we proposed a dual-microphone adaptive noise cancellation (ANC) algorithm with deferred filter coefficients for robust hotword detection in [1]. It exploits two unique hotword-related features: hotwords are the leading phrase of valid voice queries and they are short. These features allow us *not* to compute a speech-noise mask that is a common prerequisite for many multichannel speech enhancement approaches. This novel idea was found effective against strong and ambiguous speech-like TV noise. In this paper, we show that it can be generalized to support more than two microphones. The development is validated using re-recorded data with background TV noise from a 3-mic array. By adding one more microphone, the false reject (FR) rate can be further reduced relatively by 33.5%.

Index Terms: Hotword/wake-word detection, keyword spotting, multi-microphone noise cancellation, microphone array processing for machine learning

1. Introduction

Hotword (or wake word) detection is the first step to start a conversation with today's increasingly popular voice assistants like Google's Assistant and Amazon's Alexa. It is a special case of keyword spotting that listens continuously to one or multiple open microphones and recognizes a predefined hotword. Once the hotword is detected, other parts of the system will be woken up from idle to process the user's voice inputs.

While the vocabulary is extremely small, hotword detection still needs to deal with the challenges that more sophisticated automatic speech recognition (ASR) systems commonly have to overcome, including but not limited to speaker/accent variability, channel distortion, level mismatching, and noise robustness. Moreover, since this is an always-on program running on client devices, practical solutions must have low latency, small memory footprint, and light CPU load.

The research on keyword spotting can be traced back to more than four decades ago [2]. Early approaches were based on template matching via dynamic time warping [3] and probabilistic inference using hidden Markov models [4, 5, 6]. They produced promising results, but the most remarkable performance improvement in this field was driven by the advent of deep learning with artificial neural networks (NNs) [7]. Recently a variety of NN structures have been explored for hotword detection: e.g., deep neural network (DNN) [8], convolutional neural network (CNN) [9], deep residual network (ResNet) [10], recurrent neural network (RNN) [11, 12], long short-time memory (LSTM) [13, 14], and end-to-end (E2E) model [15]. These efforts have successfully brought hotword detection accuracy in clean to moderately noisy environments to a level that is acceptable for commercial deployment at scale.

By the end of 2018, according to a Voicebot study [16], over one billion devices (being either smart speakers, smartphones, or car infotainment systems) have had access to voice assistant services. This level of ubiquity and penetration implies that hotword detection may work in various acoustic environments on a daily basis, and hence demands decent robustness against strong and ambiguous multi-talker noise.

Our research at Google has investigated multi-style training (MTR) [17] which was found useful to mitigate far-field distortions (room reverberation) and non-speech noise. To address speech-like interference like TV noise, we believe that we will need to leverage multichannel speech enhancement algorithms. They may be integrated as hidden layers and jointly trained with the NN, or work as stand-alone preprocessors. Recently we proposed an ANC algorithm called hotword cleaner for dual-microphone systems [1]. It is tailored for hotword detection by exploiting two unique hotword-related features: hotwords are always the leading phrase of valid voice queries and they are short in duration. Particularly we update the cross-channel noise cancellation filter constantly like in traditional ANC systems, but the cleaner's outputs are computed using deferred filter coefficients. This simple logic is very effective on TV noise, significantly outperforming both the two-channel Wiener filter and many types of beamformers [18]. In this paper, we would like to generalize the idea to support more than 2 microphones and refer to this new algorithm as Multi-Channel Cleaner or McCleaner in short. Using re-recorded data from a 3-mic array, we will present a side-by-side comparison of the McCleaner algorithm between using 2 and 3 microphones. By adding one more microphone, the FR rate can be reduced relatively by 33.5%.

2. Signal and Array Models

In this work we use a microphone array of M elements to capture a speech source $s(t)$ in a noisy and reverberant acoustic environment. The output of the mth microphone in the time domain is expressed as

$$\begin{aligned} y_m(t) &= x_m(t) + v_m(t) \\ &= a_m * s(t) + v_m(t),\ m = 1, 2, \cdots, M, \end{aligned} \tag{1}$$

where $x_m(t)$ and $v_m(t)$ are the speech and additive noise components, respectively, a_m denotes the impulse response from the speech source to the mth microphone, and $*$ stands for linear convolution. The additive noise is the summation of the contributions from a number of Q different sound sources. Then we have

$$v_m(t) = \sum_{q=1}^{Q} b_{q,m} * u_q(t), \tag{2}$$

where $b_{q,m}$ is the impulse response from the qth noise source $u_q(t)$ to the mth microphone. Alternatively (1) can be written with respect to the signal components of each sound source at the first microphone

$$y_m(t) = h_m * x_1(t) + \sum_{q=1}^{Q} g_{q,m} * u_{q,1}(t), \tag{3}$$

where h_m represents the relative impulse response from the first to the mth microphone with respect to the speech source location, which satisfies $h_m * a_1 = a_m$, and $g_{q,m}$ is similarly defined between $b_{q,1}$ and $b_{q,m}$.

Using a K-point STFT analysis, a linear convolution is rigorously converted into a sum of K cross-band filter convolutions in the STFT domain, which are necessary to cancel the aliasing caused by downsampling in each frequency subband [19]. As a result, (3) is transformed into

$$Y_m(k,n) = \sum_{k'=0}^{K-1} \Bigg[\mathcal{H}_m(k',k) \star X_1(k',n) + \sum_{q=1}^{Q} \mathcal{G}_{q,m}(k',k) \star U_{q,1}(k',n) \Bigg], \quad (4)$$

where $Y_m(k,n)$, $X_1(k,n)$, and $U_{q,1}(k,n)$ are the STFTs of $y_m(t)$, $x_1(t)$, and $u_{q,1}(t)$, respectively, at discrete-frequency k and time frame n, $\mathcal{H}_m(k',k)$ and $\mathcal{G}_{q,m}(k',k)$ represent the cross-band (from the k'th to kth subbands) convolution filter, and $\star$ denotes linear convolution in the STFT domain along n.

When K is large, the so-called multiplicative transfer function (MTF) approximation [20] can be applied such that

$$Y_m(k,n) = H_m(k)X_1(k,n) + \sum_{q=1}^{Q} G_{q,m}(k)U_{q,1}(k,n). \quad (5)$$

But if we want to represent the long relative impulse responses with short analysis windows (smaller K), the convolutive transfer function (CTF) approximation [21] is more accurate and less restrictive, under which (4) becomes

$$Y_m(k,n) = \sum_{l=0}^{L-1} \Bigg[H_m(k,l)X_1(k,n-l) + \sum_{q=1}^{Q} G_{q,m}(k,l)U_{q,1}(k,n-l) \Bigg]. \quad (6)$$

Putting (6) in the vector/matrix form yields

$$Y_m(k,n) = \mathbf{h}_m^T(k)\mathbf{x}_1(k,n) + \sum_{q=1}^{Q} \mathbf{g}_{q,m}^T(k)\mathbf{u}_{q,1}(k), \quad (7)$$

where

$$\begin{aligned}
\mathbf{h}_m(k) &\triangleq \left[H_m(k,0) \cdots H_m(k,L-1) \right]^T, \\
\mathbf{x}_1(k,n) &\triangleq \left[X_1(k,n) \cdots X_1(k,n-L+1) \right]^T, \\
\mathbf{g}_{q,m}(k) &\triangleq \left[G_{q,m}(k,0) \cdots G_{q,m}(k,L-1) \right]^T, \\
\mathbf{u}_{q,1}(k) &\triangleq \left[U_{q,1}(k,n) \cdots U_{q,1}(k,n-L+1) \right]^T,
\end{aligned}$$

and $(\cdot)^T$ denotes the transpose of a vector or matrix.

3. Multi-Microphone Hotword McCleaner

Microphone array beamforming is probably the best-known multichannel speech enhancement method [22, 23, 18]. One of the appealing features of beamformers is that distortionless speech is theoretically deliverable: see the MVDR (minimum variance distortionless response) beamformers based on the MTF approximation [24, 18, 25] and those using the CTF approximation [21, 26]. But the emphasis on producing no to little speech distortion also limits the amount of noise that it can eliminate, particularly when the number of microphones is small. This unfortunately impairs its effectiveness for suppressing strong background noise.

While applications for voice communications care about speech quality directly, hotword detection applications are only concerned with whether audio containing the hotword, after being enhanced, triggers the detector. So we proposed in [1] a different approach called hotword cleaner, which relaxes the constraint of distortionless target speech and focuses primarily on noise cancellation. In that work only two microphones were considered ($M = 2$). Here we would like to generalize the idea to the cases of $M \geq 2$.

In order to have low latency (i.e., using a small K), the McCleaner is based on the CTF approximation and its output is given by

$$\begin{aligned}
Z(k,n) &= Y_1(k,n) - \hat{Y}_1(k,n) \\
&= Y_1(k,n) - \mathbf{w}^H(k)\mathbf{y}_{2:M}(k,n), \quad (8)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{w}(k) &\triangleq \left[\mathbf{w}_2^T(k) \cdots \mathbf{w}_M^T(k) \right]^T, \\
\mathbf{w}_m(k) &\triangleq \left[W_m(k,0) \cdots W_m(k,L-1) \right]^T, \\
\mathbf{y}_{2:M}(k,n) &\triangleq \left[\mathbf{y}_2^T(k,n) \cdots \mathbf{y}_M^T(k,n) \right]^T, \\
\mathbf{y}_m(k,n) &\triangleq \left[Y_m(k,n) \cdots Y_m(k,n-L+1) \right]^T,
\end{aligned}$$

and $(\cdot)^H$ denotes the Hermitian transpose of a vector or matrix.

We intend to find the filter coefficients $\mathbf{w}(k)$ that can cancel the noise component in the first microphone signal $Y_1(k,n)$ using the noise components in the other microphone signals $\mathbf{y}_{2:M}(k,n)$. This is accomplished by minimizing the mean squared error (MSE) of the McCleaner's output. But it is noteworthy that this minimization has to be carried out during the absence of speech, i.e., when $s(t) = 0$ or $y(t) = v(t)$. So we deduce

$$\begin{aligned}
\mathbf{w}_c(k) &= \arg\min_{\mathbf{w}(k)} E\left\{ |Z(k,n)|^2 \text{ s.t. } y(t) = v(t) \right\} \\
&= \mathbf{R}_{vv,2:M}^{-1}(k)\mathbf{r}_{v_1 v_{2:M}}(k), \quad (9)
\end{aligned}$$

where the subscript $(\cdot)_c$ stands for cleaner,

$$\begin{aligned}
\mathbf{R}_{vv,2:M}(k) &\triangleq E\left\{ \mathbf{v}_{2:M}(k,n)\mathbf{v}_{2:M}^H(k,n) \right\}, \quad (10) \\
\mathbf{r}_{v_1 v_{2:M}}(k) &\triangleq E\left\{ V_1(k,n)\mathbf{v}_{2:M}^*(k,n) \right\}, \quad (11)
\end{aligned}$$

$E\{\cdot\}$ denotes the mathematical expectation and $(\cdot)^*$ denotes the conjugate of a complex variable. The estimates of (10) and (11) are recursively updated by

$$\begin{aligned}
\hat{\mathbf{R}}_{vv,2:M}(k,n) &= \lambda\hat{\mathbf{R}}_{vv,2:M}(k,n-1) + \\
&\quad (1-\lambda)\mathbf{v}_{2:M}(k,n)\mathbf{v}_{2:M}^H(k,n), \quad (12) \\
\hat{\mathbf{r}}_{v_1 v_{2:M}}(k,n) &= \lambda\hat{\mathbf{r}}_{v_1 v_{2:M}}(k,n-1) + \\
&\quad (1-\lambda)V_1(k,n)\mathbf{v}_{2:M}^*(k,n) \quad (13)
\end{aligned}$$

where $0 < \lambda < 1$ is a forgetting factor.

Typically detecting the presence of speech relies on a voice activity detector (VAD). But when the additive noise $v(t)$ contains speech, VAD can be problematically noisy.

For the McCleaner, we can develop a smart scheme to supervise its adaptation. The scheme exploits two unique properties of hotwords: they are leading phrases of valid voice queries and have short durations. So it is designed to continuously update $\hat{\mathbf{R}}_{vv,2:M}(k,n)$ and $\hat{\mathbf{r}}_{v_1 v_{2:M}}(k,n)$ according to (12) and (13) regardless of the presence of speech. It continues to update $\mathbf{w}_c(k,n)$ at each frame

$$\mathbf{w}_c(k,n) = \hat{\mathbf{R}}_{vv,2:M}^{-1}(k,n)\hat{\mathbf{r}}_{v_1 v_{2:M}}(k,n). \qquad (14)$$

But the McCleaner's output is computed using the deferred filter coefficients instead of the filter coefficients that are just updated as follows

$$Z(k,n) = Y_1(k,n) - \mathbf{w}_c^H(k, n-d)\mathbf{y}_{2:M}(k,n), \qquad (15)$$

where d is a lag time in number of frames. In our research for detection of "Ok/Hey Google", d is set to a value with an equivalent delay of 768 ms.

Thanks to the aforementioned two unique properties of hotwords we proposed to exploit, hotwords are located in the transition periods between two acoustic scenes. So the McCleaner algorithm as formulated by (12)–(15) can be understood as an edge filter (like those in image processing) for speech: it can automatically construct a multichannel noise cancellation filter that can timely trace the ambient noise condition while the filter remains effective in the leading edge of an utterance. As a result, this tailored algorithm is simple (in the sense of easy logic) and works effectively for hotword detection.

In (14), we need to compute the inverse of $\hat{\mathbf{R}}_{vv,2:M}(k,n)$. When $M = 2$ and L is moderate, this covariance matrix is presumably non-singular. So in [1] the fast recursive least square (Fast-RLS) method was used. But when $M > 2$, $\hat{\mathbf{R}}_{vv,2:M}(k,n)$ could be singular due to inter-channel correlation. We choose to use the RLS method and compute $\hat{\mathbf{R}}_{vv,2:M}^{-1}(k,n)$ via the singular value decomposition (SVD) based pseudoinverse method.

Before we leave this section, it is important to briefly explain what we understand about the benefits of using more microphones in the McCleaner algorithm:

1) Even when there may be only one point noise source, its recordings at two microphones cannot be perfectly coherent. By adding more microphones, more reference signals are available in the signal prediction problem given by (8). While these reference signals inevitably contain redundant information, the residual error can be further minimized.
2) In many practical use cases, there can be multiple noise sources, i.e, $Q > 1$. If only two microphones are used, it is clear from (7) that the compound noise component in the first microphone cannot be possibly predicted by the noise component in the second microphone with one finite impulse response (FIR) filter. But if more microphones are employed and $M - 1 \geq Q$, perfect noise cancellation is possible according to the MINT (multichannel inverse) theory [27].

4. System Integration Strategies

When we generalize the hotword cleaner algorithm to support multiple microphones, the system integration strategies discussed in [1] are still relevant and need only minor modification. For self-containedness, Fig. 1 updates the three strategies for 3 microphones. But for brevity of presentation, we will make no further discussion on this. It is simply worth reiterating that the cleaner-only strategy can reduce false alarms and is more computationally efficient while the hybrid strategy minimizes latency and can save some of McCleaner's corner cases in quiet conditions.

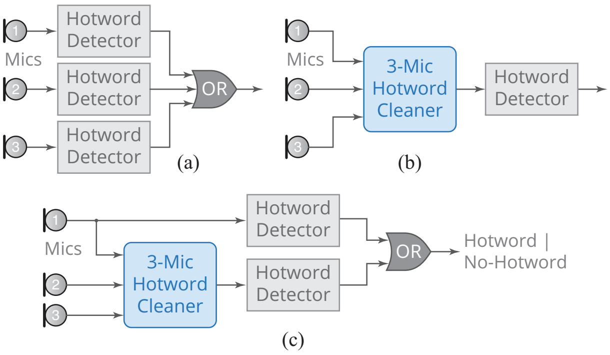

Figure 1: *Three strategies for integrating 3-mic hotword cleaner and detector: (a) baseline, (b) cleaner-only, and (c) hybrid.*

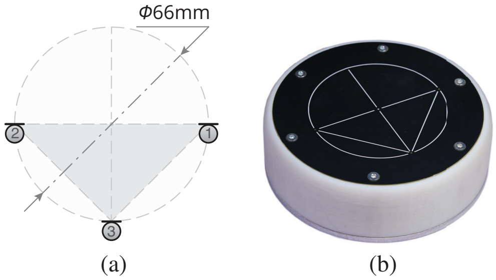

Figure 2: *The right isosceles triangular microphone array used for data collection: (a) array topology and (b) circuit board.*

5. Experiments

5.1. Microphone Array and Data Collection

In our research, we collected re-recorded audio data to prove the concept that hotword McCleaner can work better by using more microphones. A triangular array was developed as shown in Fig. 2. It consists of three microphones on a circle of diameter 66 mm. Recordings took place in a 4.6 m × 4.7 m × 2.7 m living-room lab with no sound treatment. We considered only two acoustic conditions: far-field clean and TV noise in the background. The TV was placed against a wall. Clean speech utterances were played back from a mouth simulator (Brüel & Kjær TYPE 4227), which was positioned about 2.7 m in front of the TV. The microphone array was mounted at two different locations: horizontally on a table beneath the TV and vertically on the wall behind the TV. At each position at least 2,000 utterances were recorded, all with long preambles (longer than 10 s). Table 1 presents a summary of the data. The negative dataset is composed of TV audio recorded on a far-field Google Home in previous studies. All utterances are sampled at 16 kHz with 16 bits.

5.2. Experimental Setup

For the results presented below, we have chosen the following parameters for the hotword McCleaner regardless of how many microphones it uses: 128 ms STFT windows with 50% overlap, $d = 12$, $L = 3$, $\lambda = 0.99$. The covariance matrix in (12) is initialized by $\hat{\mathbf{R}}_{vv,2:M}(k,0) = \delta\mathbf{I}$, where $\delta = 10^{-4}$. Matrix pseudoinverse is computed via the JacobiSVD method from the Eigen library in C++ [28]: the method's default value is used for the threshold below which singular values are considered as zeros.

The end-to-end (E2E) model for hotword detection [15] was implemented using Google's TensorFlow™ library [29]

Table 1: *Summary of collected data for performance evaluation.*

Dataset	Num. Utterances	Length (hours)
Far-Field Clean	4,507	22.69
TV Noise	4,307	22.29
Negative	55,469	1,175.09

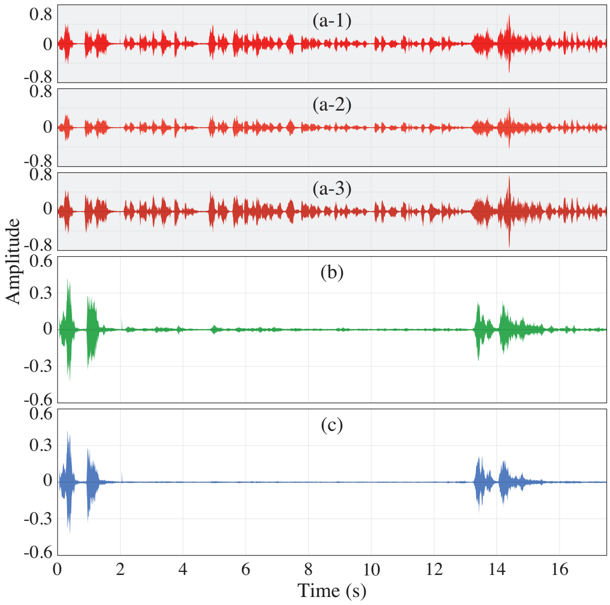

Figure 3: *Sample multichannel utterance collected from the isosceles triangular microphone array mounted on a wall with TV noise and the processed signals by using the hotword McCleaner: (a) the three microphone signals, (b) the cleaned signal using the first two microphones, and (c) the cleaned signal using all three microphones.*

and trained with deep learning algorithms on both logs and collected data from gender-balanced pool of volunteers with a variety of accents [30].

5.3. Cleaned Sample Utterance

Before we present an objective validation of hotword McCleaner's performance, it is helpful to examine some sample utterances processed by the McCleaner using both 2 and 3 microphones. This subjective comparison can shed some insight on how much improvement performance-wise we can expect by adding one more microphone. Figure 3 visualizes one of such samples. The utterance has 3 audio channels and lasts for 17.5 s. The hotword followed by a query appears around 13 s and is heavily contaminated by TV noise. The SNR is about 0 dB. The cleaned signals using the first 2 and all 3 input channels are plotted in Fig. 3(b) and (c), respectively. While the 2-mic cleaner already does a good job in noise suppression, the 3-mic algorithm makes the residual noise perceivably much lower and smoother. This helps recognize the last consonant `/l/` in ”Ok/Hey Google”, which is often pronounced weakly by nature. In both cases the cleaned hotword gets attenuated too. But the SNR and speech quality have significantly been improved.

5.4. Receiver Operating Curves (ROCs)

We choose ROC, the most informative performance measure, to evaluate and compare hotword detection systems. A ROC plots the FR rate (likelihood of an FR per hotword-containing instance) as a function of the FA rate (in number of FAs per hour of negative audio). Figure 4 presents the ROCs of 5 systems under test. The baseline follows what was illustrated by Fig. 1(a) and uses all the 3 input audio channels. The other cleaner-only and hybrid systems follow the strategies of Fig. 1(b) and (c), respectively, but using different number of microphones.

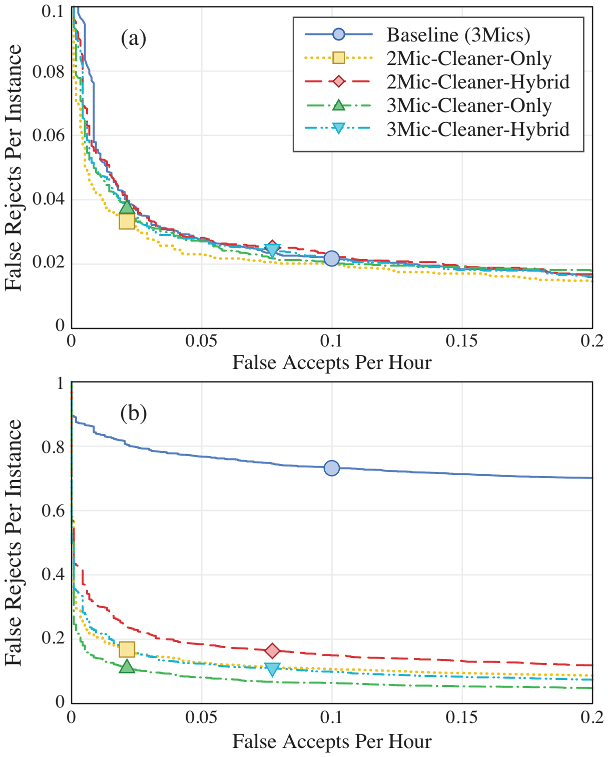

Figure 4: *ROCs comparing performance of the hotword McCleaner using 2 and 3 microphones against the baseline system on (a) far-field clean and (b) TV-noise datasets. Note that the marker positions correspond to the same threshold of the E2E hotword detection model.*

On the far-field clean dataset, the performance of these systems are on a similar level: the 2mic-cleaner-only is slightly better than the others. On the TV-noise dataset, the McCleaner systems decisively outperform the baseline. The FR rate at the operating point where the detector's threshold is fixed is reduced from about 75% to lower than 17%. Among the McCleaner systems, using 3 microphones is more advantageous than using 2 microphones with a clear margin. At the operating point, the hybrid systems' FR rates by using 2 and 3 microphones are 16.4% and 10.9%, respectively. By adding one more microphone, we achieved a 33.5% relative reduction in FR rate.

6. Conclusions

In this paper we have generalized the speech enhancement algorithm by adaptive noise cancellation with deferred filter coefficients for robust hotword detection from dual to multi-microphone systems. To justify the development, we collected re-recorded data with strong TV noise in the background from a 3-mic array. We have presented a comparison of the proposed algorithm between using 2 and 3 microphones. It was shown that by adding one more microphone, the FR rate can be further reduced relatively by 33.5%.

7. References

[1] Y. Huang, T. Z. Shabestary, and A. Gruenstein, “Hotword cleaner: dual-microphone adaptive noise cancellation with deferred filter coefficients for robust keyword spotting,” in *Proc. IEEE ICASSP*, 2019, pp. 6346–6350.

[2] J. S. Bridle, “An efficient elastic-template method for detecting given words in running speech,” *Brit. Acoust. Soc. Meeting*, pp. 1–4, Apr. 1973.

[3] A. L. Higgins and R. E. Wohlford, “Keyword recognition using template concatenation,” in *Proc. IEEE ICASSP*, 1985, pp. 1233–1236.

[4] R. C. Rose and D. B. Paul, “A hidden Markov model based keyword recognition system,” in *Proc. IEEE ICASSP*, 1990, pp. 129–132.

[5] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, “Continuous hidden Markov modeling for speaker-independent wordspotting,” in *Proc. IEEE ICASSP*, 1990, pp. 627–630.

[6] J. G. Wilpon, L. G. Miller, and P. Modi, “Improvements and applications for key word recognition using hidden Markov modeling techniques,” in *Proc. IEEE ICASSP*, 1991, pp. 309–312.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, September 2012.

[8] G. Chen, C. Parada, and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” in *Proc. IEEE ICASSP*, 2014, pp. 4087–4091.

[9] T. N. Sainath and C. Parada, “Convolutional neural networks for small-footprint keyword spotting,” in *Proc. InterSpeech*, 2015, pp. 1478–1482.

[10] R. Tang and J. Lin, “Deep residual learning for small-footprint keyword spotting,” in *Proc. IEEE ICASSP*, 2018, pp. 5484–5488.

[11] S. Fernández, A. Graves, and J. Schmidhuber, “An application of recurrent neural networks to discriminative keyword spotting,” in *Proc. Int. Conf. Artificial Neural Networks*, 2007, pp. 220–229.

[12] P. Baljekar, J. F. Lehman, and R. Singh, “Online word-spotting in continuous speech with recurrent neural networks,” in *Proc. Spoken Language Technology Workshop (SLT)*, 2014, pp. 536–541.

[13] M. Wöllmer, B. Schuller, and G. Rigoll, “Keyword spotting exploiting long short-term memory,” *Speech Commun.*, vol. 55, pp. 252–265, February 2013.

[14] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni, “Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting,” in *Proc. Spoken Language Technology Workshop (SLT)*, 2016, pp. 474–480.

[15] R. Alvarez and H.-J. Park, “End-to-end streaming keyword spotting,” IEEE ICASSP, pp. 6336–6340, 2019.

[16] Voicebot.ai, “Voice assistant consumer adoption report,” https://voicebot.ai/wp-content/uploads/2018/11/voice-assistant-consumer-adoption-report-2018-voicebot.pdf, Nov. 2018, [Online; Latest Accessed 24-Feb-2019].

[17] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, “Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” in *Proc. InterSpeech*, 2017, pp. 379–383.

[18] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin, Germany: Springer, 2008.

[19] M. Portnof, “Time frequency representation of digital signals and systems based on short-time Fourier analysis,” *IEEE Trans. Signal Process.*, vol. ASSP-28, pp. 55–69, Feb. 1980.

[20] Y. Avargel and I. Cohen, “On multiplicative transfer function approximation in the short time Fourier transform domain,” *IEEE Signal Process. Lett.*, vol. 14, pp. 337–340, May. 2007.

[21] R. Talmon, I. Cohen, and S. Gannot, “Relative transfer function identification using convolutive transfer function approximation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, pp. 546–555, May. 2008.

[22] B. Van Veen and K. Buckley, “Beamforming: A versatile approach to spatial filtering,” *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1998.

[23] M. Brandstein and D. Ward, Eds., *Microphone Arrays*. Berlin, Germany: Springer-Verlag, 2001.

[24] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Trans. Signal Process.*, vol. 49, pp. 1614–1626, Aug. 2001.

[25] M. Souden, J. Benesty, and S. Affes, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 260–276, Feb. 2010.

[26] E. A. P. Habets, J. Benesty, and J. Chen, “Multi-microphone noise reduction using interchannel and interframe correlations,” in *Proc. IEEE ICASSP*, 2012, pp. 305–308.

[27] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-36, pp. 145–152, Feb. 1988.

[28] G. Guennebaud, B. Jacob *et al.*, “Eigen: An open-source high-level C++ library for linear algebra,” http://eigen.tuxfamily.org, [Online; Latest Accessed 15-Mar-2019].

[29] Google Inc., “TensorFlow™: An end-to-end open source machine learning platform,” https://www.tensorflow.org, [Online; Latest Accessed 15-Mar-2019].

[30] T. Hughes, K. Nakajima, L. Ha, A. Vasu, P. Moreno, and M. LeBeau, “Building transcribed speech corpora quickly and cheaply for many languages,” in *Proc. InterSpeech*, 2010, pp. 1914–1917.