THE NTU-ADSC SYSTEMS FOR REVERBERATION CHALLENGE 2014

Xiong Xiao[1], *Shengkui Zhao*[2], *Duc Hoang Ha Nguyen*[3], *Xionghu Zhong*[3], *Douglas L. Jones*[2], *Eng Siong Chng*[1,3], *Haizhou Li*[1,3,4]

[1]Temasek Lab@NTU, Nanyang Technological University, Singapore
[2]Advanced Digital Sciences Center, Singapore
[3]School of Computer Engineering, Nanyang Technological University, Singapore
[4]Department of Human Language Technology, Institute for Infocomm Research, Singapore

xiaoxiong@ntu.edu.sg, shengkui.zhao@adsc.com.sg, ng0008ha@ntu.edu.sg, xhzhong@ntu.edu.sg

ABSTRACT

This paper describes our speech enhancement and recognition systems developed for the Reverberation Challenge 2014. To enhance the noisy and reverberant speech for human listening, besides using conventional methods such as delay and sum beamformer and late reverberation reduction by spectral subtraction, we also studied a novel learning-based speech enhancement. Specifically, we train deep neural networks (DNN) to map reverberant spectrogram to the corresponding clean spectrogram by using parallel data of clean and reverberant speech. Results show that the trained DNN is able to reduce reverberation significantly for unseen test data.

For the speech recognition task, when parallel data is available, we train a DNN to map reverberant features to clean features, following the same spirit as the DNN-based speech enhancement. Results show that the DNN-based feature compensation improves speech recognition performance even when a DNN acoustic model is already used, showing the benefit of explicitly cleansing the features. When parallel data is not available in the clean condition training scheme, we focus on reducing the training-test mismatch by using our proposed cross transform feature adaptation that uses both temporal and spectral information. The cross transform works complementarily with traditional model adaptation.

Index Terms— speech enhancement, beamforming, robust speech recognition, feature compensation, reverberation challenge, feature adaptation, deep neural networks.

1. INTRODUCTION

Automatic speech recognition (ASR) systems have achieved good performance for speech collected by close-talking microphones. However, recent development in speech and audio applications such as multimedia, hearing aids, and hands-free speech communication systems require speech acquisition in a distant-talking environment. Unfortunately, as the distance between the mouth to the microphone increases, the recorded speech become increasingly distorted due to background noise and room reverberation. Consequently, ASR performance can be significantly degraded.

Recently, Reverberation Challenge 2014 [1] was introduced as a common benchmark to evaluate dereverberation techniques for both human listening and ASR. The distortion considered is mainly reverberation, with moderate amount of additive background noise. This paper describes the systems we built for the challenge.

Reverberation is produced by multi-path propagation of an acoustic signal from its source to the microphone. The distortion can be described by acoustic impulse response (AIR) that may last for hundreds of milliseconds. Reverberation cancelation has been tried by many researches using the deconvolution techniques that estimate and apply the inverse of AIR [2–8]. However, the available techniques are sensitive to estimation error of AIR, which is difficult to estimate in realistic environment.

Reverberation suppression using spatial processing and spectral enhancement are found to be more practical and robust. Microphone array processing techniques provide a spatial filtering to suppress specular reflections so that the speech signal from the desired direction of arrival (DOA) can be enhanced. In general, adaptive beamformers are more preferred in both denoising and interference suppression than fixed beamformers in case of independent distortions [9–13]. Unfortunately, reverberant speech signals consist of highly dependent distortions that are delayed versions of the signal itself. Hence, many traditional adaptive beamformers become ineffective in reverberation suppression. Very recently, a conjunction of delay-and-sum (DS) beamformer and minimum variance distortionless response (MVDR) beamformer shows favorable performance for the speech enhancement in the room environment [14].

In this work, we employ conventional beamformers, such as DS and MVDR, to take advantage of multiple microphone recordings using spatial filtering. We also use spectral subtraction to further attenuate late reverberation, as suggested in [15, 16]. Besides these conventional methods, we investigated a different approach for speech dereverberation that is based on learning from data. Specifically, we let deep neural networks (DNN) learn how to map reverberant spectrogram to their clean version from clean and reverberant spectrogram pairs. If given enough training samples, we expect DNN to be able to dereverberate unseen test utterances that is not too different from the training data. Such concept is also applied to estimated clean features from reverberant features for speech recognition task. When no parallel clean and reverberant data is available in the clean condition training scheme, we apply a novel feature adaptation method that uses both speech spectral and temporal information to reduce training-test mismatch.

The rest of the paper is organized as follows. Section 2 presents the corpus and tasks in the reverberation challenge 2014. Section 3 describes the key technologies in the speech enhancement and recognition systems. In section 4, experimental results are presented and discussed. Finally, we conclude in section 5.

2. CORPUS AND TASKS

In this reverberation challenge, there are 2 types of data to evaluate speech enhancement and robust speech recognition techniques. One

type is simulated reverberant and noisy speech, generated by adding noise and reverbeartion to clean utterances from WSJCAM0 [17] corpus, which is the British version of the WSJ0 corpus [18]. The other type is real meeting recording from the MC-WSJ-AV [19] corpus, which is the re-recorded version of WSJCAM0 in a meeting room environment. A development set containing these two types of data are provided to participating teams to tune their systems before the evaluating period. Another evaluation set, which is similar to the development set in terms of distortion characteristics, is used for evaluation.

In the speech recognition task, there are two training schemes, i.e. 1) the clean condition scheme in which only clean data is available for training the acoustic model; 2) multi condition training scheme in which reverberant and noisy speech with similar characteristics as the simulated test data are available for training. The clean condition training data are taken from the WSJCAM0 [17] corpus, while the multi condition training data is artificially generated by corrupting clean condition training data in similar way as the generation of simulated test data.

Speech enhancement methods are evaluated by several metrics, such as cepstral distance (CD) [20], log likelihood ratio (LLR) [20], frequency-weighted segmental SNR [20], Speech-to-reverberation modulation energy ratio (SRMR) [21], and optional PESQ [22]. For real rooms data, only the non-intrusive SRMR metric is used. In addition, subjective listening is planned by the organizer of the challenge. The ASR system is evaluated by word error rate (WER). As we don't have the PESQ license, we will not report PESQ results in this paper. For more details of the corpus and tasks, please visit the Reverberation Challenge website (http://reverb2014.dereverberation.com).

3. SYSTEM DESCRIPTION

3.1. Speech Enhancement Systems

We developed two speech enhancement systems as illustrated in Fig. 1. The first system uses DS beamforming followed by spectral subtraction for removing late reverberations. The second system uses MVDR beamforming followed by DNN-based spectrogram enhancement.

3.1.1. DS beamformer plus spectral subtraction

The block diagram of this speech enhancement system is presented in Fig. 2. The DS beamformer was developed to increase the output signal to noise ratio (SNR) [9]. It exploits the fact that the time of arrival (TOA) of an incoming signal at the different microphones are different. It is a signal independent approach performed by aligning multi-channel signals according to the corresponding time-delay of arrivals (TDOAs). By summing the microphone outputs in phase of the desired direction, the speech frequency components of the desired direction reinforce while the frequency components of the off-desired direction cancel. In order to match the phases of speech signals from the desired direction, we compute the TDOAs using the GCC-PHAT method [23] with a Hamming window of 64 ms long. The first channel in each record was selected as the reference channel. The TDOAs between the reference channel and the other channels were calculated from the cross-correlation peaks that appeared within the maximum delay of 0.59 ms. The maximum delay was computed from the diameter of 20 cm of the circular microphone array and the speed of sound of 340 m/s. In the DS beamformer, the same window size of 64 ms was used with 75% overlap. We

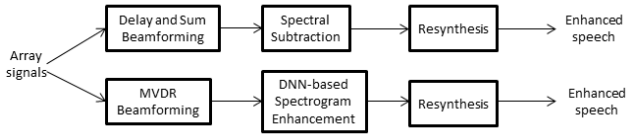

Fig. 1. Block diagram of the two speech enhancement systems.

applied the pre-emphasis and the Hanning window before taking the short-time Fourier transform (STFT) of 1024 points. The frequency components of all the channels were aligned using the computed TDOAs and summed with a normalization to obtain the frequency components of output. Then we apply the inverse STFT and the overlap-save method to obtain the time-domain signal of 16 ms long. The de-emphasis was performed to obtain the final output of the DS beamformer. Then the output of the DS beamformer is passed to the spectral subtraction module.

The DS beamformer is effective for attenuating the reverberation that are reflected from the directions that are different from the desired direction. However, the DS beamformer tends to attenuate less when the reflected paths are close to the direct path. In this case, the desired speech signals still consist of the reflected signals. The spectral subtraction approach [15] focused on the overlap-masking that is the energy of the current signal $x(t)$ overlaps the following signal $x(t+T)$. Considering the exponential model of the room impulse response and the quasi-stationary characteristics of speech signals, we used the following formulation for amplitude spectral subtraction [15]:

$$|\hat{S}(m,k)| = \Big(1 - \frac{1}{\sqrt{\text{SNR}_{pri}+1}}\Big)|X(m,k)|, \qquad (1)$$

where $\hat{S}(m,k)$ is the estimate of the amplitude spectrum of the dereverberated signal at the frame index m and the frequency index k, and $X(m,k)$ is the amplitude spectrum of the reverberated output signal of the DS beamformer. We transformed the time-domain output into the frequency domain using the STFT of 256 points after applying the pre-emphasis and the Hanning window of 16 ms long with 75% overlap. The term SNR_{pri} in (1) is estimated as

$$\text{SNR}_{pri}(m,k) = \beta\text{SNR}_{pri}(m-1,k) + (1-\beta)\text{max}[0, \text{SNR}_{pos}],$$

where β is a smoothing factor and it was set to $\beta = 0.9$ in our experiments, and the term SNR_{pos} is defined as $\text{SNR}_{pos} = \frac{|X(m,k)|^2}{\hat{\gamma}_{rr}(m,k)} - 1$. The estimated power spectral density (PSD), $\hat{\gamma}_{rr}(m,k)$, of the reverberated part of the signal $X(m,k)$ was estimated using $\hat{\gamma}_{rr}(m,k) = e^{-2\delta T}\hat{\gamma}_{xx}(m-T,k)$, where $\hat{\gamma}_{xx}(m-T,k)$ denotes the PSD of the past signal, δ is linked to the reverberation time T_r through $\delta = \frac{3\ln 10}{T_r}$, and T was set to: $T \simeq 50$ ms which is equivalent to 3 frames. The PSD of the signal is estimated as $\hat{\gamma}_{xx}(m,k) = \alpha\hat{\gamma}_{xx}(m-1,k) + (1-\alpha)|X(m,k)|^2$, where the smoothing factor α was set to 0.7 in our experiments.

There are many approaches proposed to estimate the reverberation time T_r in the literature. A recent maximum-likelihood (ML) estimator presented in [24] shows improved performance with lower computational complexity. Therefore, we implemented the estimator to estimate the reverberation time using the output of the DS beamformer. Noted that the estimator in [24] takes the advantage of a long input signal, we repeatedly refined the T_r estimation by accumulating the output signals of the DS beamformer for each utterance.

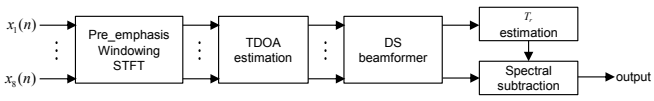

Fig. 2. Block diagram of the speech enhancement system using the DS beamformer and spectral subtraction.

3.1.2. MVDR beamformer

Beside the speech enhancement system described above, we also investigate the use of DNN for dereverberation. In this and next section, we describe our second speech enhancement system using the MVDR beamformer and DNN. The MVDR beamformer is used to perform spatial filtering on the reverberant speech signals of the training and evaluation data. DNN is trained to map the spectrogram of the MVDR-processed signal to the underlying clean speech spectrogram. We will describe MVDR in this section and DNN-based spectrogram enhancement in the next section.

We choose to use the MVDR beamformer due to its better capability for maximizing the output SNR as compared to the DS beamformer from our results on the given evaluation data. In our implementation, we used the window size of 64 ms with 75% overlap which is the same as in the DS beamformer. The signal was first pre-emphasized and then the Hamming window was applied. The STFT of 1024 points was used to transformed the time-domain signal into the frequency domain. The MVDR beamformer is slightly more complicated than the DS beamformer due to the need of noise covariance matrix in the estimation of weight vector. The statistical solution for computing the weight vector of the MVDR beamformer is expressed below [25]:

$$\mathbf{w}_{MVDR}(m,k) = \frac{\mathbf{R}_{\mathbf{nn}}^{-\mathbf{1}}(m,k)\mathbf{e}(k)}{\mathbf{e}^H(k)\mathbf{R}_{\mathbf{nn}}^{-\mathbf{1}}(m,k)\mathbf{e}(k)}, \quad (2)$$

where $\mathbf{R_{nn}}(m,k)$ denotes the noise covariance matrix, which is estimated at the time index m and the frequency index k, $\mathbf{e}(k)$ represents the frequency-domain array manifold given the direction-of-arrival (DOA) of the desired speech source. To obtain the array manifold of the desired speech source in each utterance, we compute the TDOAs, $\tau_{0,i}$, $(i = 1, 2, \ldots, 7)$ using the GCC-PHAT method again by choosing the first channel as the reference channel 0. Then the array manifold can be expressed as

$$\mathbf{e}(k) = [1,\ e^{-j2\pi k F_s \tau_{0,1}/K},\ \cdots, e^{-j2\pi k F_s \tau_{0,7}/K}], \quad (3)$$

where F_s is the sampling rate, and $K = 1024$ is the length of the STFT. Once the array manifold is computed, we next need to estimate the noise covariance matrix $\mathbf{R_{nn}}$. Noted that for the case the noise and interference signals are independent to the desired signal, the noise covariance matrix can be replaced by the signal-plus-noise covariance matrix. In this challenge, since we are dealing with the reverberant signals that are dependent on the desired signal, the noise covariance matrix is used to avoid high signal distortion. A simple energy based voice activity detector (VAD) was used to detect the presence of speech. When the speech is absent, the noise covariance matrix is updated as follows:

$$\mathbf{R_{nn}}(m,k) = \lambda \mathbf{R_{nn}}(m-1,k) + (1-\lambda)\mathbf{x}(m,k)\mathbf{x}^H(m,k), \quad (4)$$

where λ is a smooth factor which was set to 0.98, $\mathbf{x}(m,k)$ is the frequency signal vector of all the channels. In our implementation, $\mathbf{R_{nn}}(m,k)$ was initialized as a diagonal matrix with equal small values for the diagonal elements.

The frequency-domain output of the MVDR beamformer is obtained as $y(m,k) = \mathbf{w}_{MVDR}^H(m,k)\mathbf{x}(m,k)$, which will be further enhanced by the DNN spectrogram enhancement that will described in the following section.

3.1.3. DNN based spectrogram enhancement

Neural networks (NN) are universal mapping functions that could be used for both classification and regression problems. NN has been used for speech enhancement for a long time [26]. An NN with more than 1 hidden layers is usually called a deep NN, or DNN. Recently, DNN becomes popular after a pretraining step, called restrictive Boltzmann machine (RBM) pretraining [27, 28], was introduced to initialize the network parameters to some reasonable values such that backpropagation can be then used to train the network efficiently on task dependent objective functions. The advantage of DNN over one-hidden-layer NN is that the deep structure of DNN allows much more efficient representation of many nonlinear transformations/functions [27]. In the past several years, DNN has been applied to many speech processing tasks, such as acoustic modeling [29] and speech enhancement (denoising) [30]. For the reverberation challenge, we applied DNN to enhance the spectrogram of distorted speech or MVDR beamformer's output. The motivation is to harness the flexibility of DNN to model the highly nonlinear and complicated mapping from distorted spectrogram to the underlying clean spectrogram. Note that we also applied DNN to map distorted MFCC features to clean features and their basic concepts are the same, we will use a unified description for both tasks here.

The structure of the DNN used for speech enhancement and ASR feature compensation is shown in Fig. 3. At the bottom of the figure there is a sequence of feature vectors generated from the noisy and reverberant speech. For speech enhancement, we used 257-dimensional log spectrum as the feature vectors, while for ASR feature compensation, we used the 39-dimensional MFCC as feature vectors. For speech enhancement, we also applied cepstral mean normalization (CMN) to the log spectrogram in an utterance-based manner to reduce any channel variations. For feature compensation, we applied mean and variance normalization (MVN) on MFCC features as we found that MVN is a better feature normalization for the speech recognition task. To predict the clean feature vector of the current frame (shown in gray color in the figure), a sequence of feature vectors around the current frame are fed into the DNN. This allows the DNN to use context information to predict the clean feature vector and is believed to be especially important for dereverberation as the effect of reverberation can last for dozens of frames. After nonlinear transformation by the hidden layers, a linear output layer is used to predict the clean feature vectors for current frame.

To train DNN for spectrogram enhancement or feature compensation, parallel data consisting of clean and distorted version of the same utterance is needed. The clean and distorted spectrograms or MFCC feature vector sequences must be aligned accurately in frame level. In the reverberation challenge, we use the clean and multi condition training data for the training of the DNN. The objective of the training is to minimize the mean square error (MSE) between the output of the DNN and the corresponding clean spectrum or MFCC features. Before the MSE training, the DNN is initialized by RBM pretraining, which is an unsupervised learning and does not require any label of the training data. The RBM training only requires the distorted version of the parallel data. Once the DNN is trained, it is expected to handle well unseen test speech utterances, whose distor-

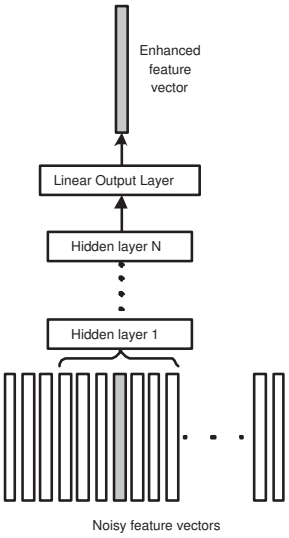

Fig. 3. Structure of DNN for spectrogram dereverberation and feature compensation.

tion characteristics are similar to those in the training data.

A drawback of the described enhancement scheme is that each frame is predicted independently and we cannot guarantee that the predicted frame sequence is smooth and sounds natural. Smoothness is not a big issue for the speech recognition task, however, it is important for the enhanced speech to sound natural. Hence, we also investigated another scheme, in which we train the DNN to predict clean spectrogram, together with its time derivatives. Similar to time derivative features in speech recognition, we used both delta spectrum (first order time derivative) and acceleration spectrum (second order time derivative). The delta and acceleration spectra vectors are concatenated to the original spectrum vector (called static spectrum) to form the new target vector for DNN learning. Hence, the target vector's dimension becomes 3 times of the original one. During enhancement phase, both the static, delta, and acceleration spectrum are predicted. The final enhanced spectrogram can be found by using a linear square fitting which minimizes $||\mathbf{X} - \mathbf{Y}||_F^2$, where $\mathbf{Y}$ is the output spectrogram matrix of DNN and contains both static, delta, and acceleration spectra, $\mathbf{X}$ is the spectrogram generated from the unknown final static spectrogram using the delta and acceleration generation formula [31]. Closed form solution can be found for the MSE problem and the static portion of $\mathbf{X}$ will be treated as the final output of the enhancement system. With such a scheme, the delta and acceleration features can help to make the static features more smooth and natural. As the dynamic range of the delta and acceleration spectra are much smaller than that of static spectrum, the weights of them in the MSE function are boosted by 3 and 5 times, respectively through empirical analysis.

3.2. Speech Recognition Systems

In Reverberation Challenge 2014, there are two speech recognition training schemes, i.e. the clean and multi condition schemes, which are different in several aspects. In clean condition training scheme, as the mismatch between the noisy and reverberant test data and the clean training data is large, the most important issue is to reduce the mismatch, e.g. by feature or model adaptation methods. Discriminative training methods or models are less effective in this case as what is learnt from clean data may not work well for very mismatched test data. On the other hand, in multi condition training, the mismatch between training and testing data are much smaller than that in clean condition training. Hence, feature or model adaptation methods become less critical, but discriminative methods becomes useful as the discriminative features or models learnt from training data are expected to also work well on the test data.

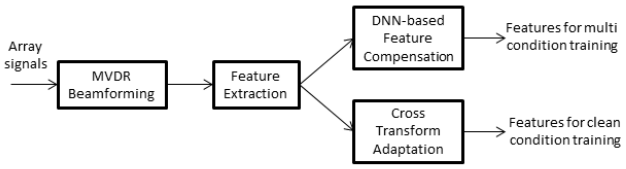

Fig. 4. Block diagram of feature processing for speech recognition systems.

The organizer of the challenge provided a simple HTK-based speech recognition system for the purpose of mainly evaluating feature domain techniques. In our preliminary study, we found that the HTK baseline system, which is based on conventional HMM/GMM model, is a good benchmark for the clean condition training scheme, but is too weak for the multi condition training scheme. For example, using DNN based acoustic model can outperform the HTK baseline significantly when multi condition training scheme is used. As a result, it is not very meaningful to test our techniques on the HTK baseline using multi condition training scheme. Therefore, in this paper, we will always use the HTK baseline system for evaluation using clean condition training, but use our own DNN based system for evaluation using multi condition training.

The feature processing for clean and multi condition training schemes are illustrated in Fig. 4. MVDR beamforming is always used when more than 1 microphone is available. In addition, we studied two feature processing techniques to enhance the noisy and reverberation features. One is the DNN based feature compensation method used in multi condition training scheme, and the other is the cross transform method to be used for clean condition training. The DNN based feature compensation is similar to the DNN based speech enhancement described previously. The major difference is that for feature compensation, MFCC features are compensated as they are directly used for speech recognition. Hence, we will not describe it again in this section.

3.2.1. Cross Transform Feature Adaptation

To compensate speech features for robust ASR, there are two popular feature processing schemes, i.e. the linear transformation of feature vectors and the temporal filtering of feature trajectories, as illustrated in Fig. 5. Linear transformation uses all dimensions of the current frame to predict new features that fit the acoustic model under maximum likelihood (ML) criterion [32,33]. On the other hand, temporal filtering uses the context information in neighboring frames to estimate features that fit the acoustic model [34–37]. While linear transformation uses inter-dimensional correlation information (or spectral information) to process features, temporal filtering uses inter-frame correlation information (or temporal information). In the past, these two types of information are usually not used together for feature adaptation. In this section, we apply our recently proposed feature adaptation method, called cross transform [38], to the speech recog-

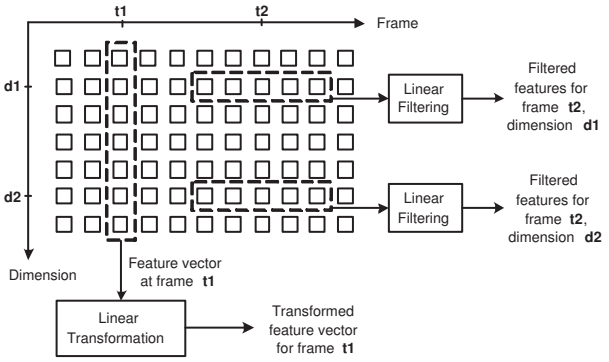

Fig. 5. Temporal filtering of feature trajectories vs. linear transformation of feature vectors.

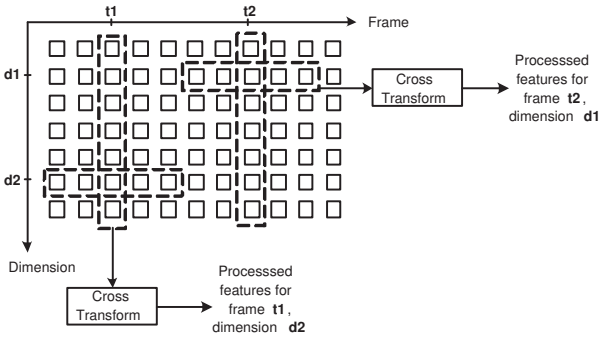

Fig. 6. Illustration of cross transform.

nition task of the reverberation challenge. For the completeness of the paper, we will briefly describe the concept of the cross transform in the following text.

To use both spectral and temporal information for feature processing, the simplest way is to predict the clean feature vectors from a sequence of input feature vectors as follows

$$\mathbf{y}_t = \sum_{\tau=-L}^{L} \mathbf{B}_\tau \mathbf{x}_{t+\tau} + \mathbf{c} = \mathbf{W}\tilde{\mathbf{x}}, \tag{5}$$

where $\mathbf{x}_t$ and $\mathbf{y}_t$ are D dimensional input and output feature vectors, respectively. $\mathbf{B}_\tau,\ \tau = -L, ..., L$ are the transformation matrices, and $\mathbf{W} = [\mathbf{B}_{-L}, ..., \mathbf{B}_L, \mathbf{c}]$ and $\tilde{\mathbf{x}}_t = [\mathbf{x}_{t-L}^T, ..., \mathbf{x}_{t+L}^T, 1]^T$ are the concatenated transformation matrices and inputs, respectively. Although the transform in (5) is possible in theory, it is hard to be applied in practice as there are too many parameters in $\mathbf{W}$ and hence a lot of data are required for its reliable estimation. For example, if we set $L = 16$, i.e. use a context of 33 frames, then there are $33D^2 + D$ parameters, which is not feasible to be reliably estimated from a small amount of test data, e.g. one test utterance. Therefore, in this study, we make $\mathbf{W}$ sparse by setting most of its elements to zero. Specifically, to predict the feature at frame t and dimension d, $y_t^{(d)}$, we only use the local feature trajectory and feature vector that contains $x_t^{(d)}$ as shown in Fig. 6. The simplified transform is simply the combination of the linear transform and temporal filter illustrated in Fig. 5. As the shape of the transform often looks like a cross, we will call it cross transform.

Similar to maximum normalized likelihood linear filtering in [37], the parameters of the cross transform can be estimated by minimizing an approximated KL divergence between the distribution of processed features, p_y, and the distribution of clean training features, p_Λ. In this work, p_y is modelled by a single Gaussian and p_Λ by a GMM with parameter set $\Lambda = \{c_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m | m = 1, ..., M\}$ and $\boldsymbol{\Sigma}_m$ being diagonal. The optimal $\mathbf{W}$ is found by minimizing the following approximated KL divergence:

$$f(\mathbf{W}) = const - \frac{\lambda}{2}\log\det(\mathbf{W}\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}^T) + \frac{\beta}{2T}||\mathbf{W} - \mathbf{W}_0||_2 - \frac{1}{T}\sum_{t=1}^{T}\log\sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{W}\tilde{\mathbf{x}}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \tag{6}$$

where $\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}}$ is the covariance matrix of $\tilde{\mathbf{x}}$. Tunable parameters β and λ are used to control the contributions of the L2 norm term and log determinant term in the cost function, respectively. An intuitive explanation of the cost function is that we want the likelihood of the transformed features on the clean reference model Λ to be high. At the same time, we want the log determinant of the covariance matrix of the transformed features to be big to prevent the variance of transformation features from shrinking too much. The cost function can be minimized via an EM algorithm iteratively. For detailed solution of the cross transform, readers are referred to [38].

4. EXPERIMENTS

4.1. Speech Enhancement

4.1.1. DS beamforming and spectral subtraction

The speech enhancement system using the DS beamformer and spectral subtraction was implemented in real-time processing scheme. As discussed in Section 3.1.1, our algorithm uses a window length of 64 ms with 75% overlap. Therefore, the actual time delay is 64 ms. We performed the speech enhancement system evaluation on the evaluation data for all the channel types of 1ch, 2ch, and 8ch. Notice that when we performed the system for the case of 1ch, the TDOA estimation and the DS beamformer were not used. The input signals were directly sent to the spectral subtraction module. The detailed results for the all the 3 channel types are shown in Table 1,2, and 3.

There are three major observations from the results. First, both the DS beamformer and spectral subtraction improve the quality of the speech signals for all the evaluation measures. Second, the combination of the DS beamformer and spectral subtraction generally works better than the spectral subtraction alone. Third, the results using 8 channels are much better than the results using only 1 or 2 channels. It is because the performance of the DS beamformer directly relies on the number of channels.

Using the provided CPU clock measuring function, our method is about 5.5 times slower than the reference code.

4.1.2. MVDR beamforming and DNN-based enhancement

We built 5 DNN-based spectrogram enhancement systems as shown in Table 4. In DNN1, the input layer takes 11 frames of spectrum (i.e. about 110 ms), so the input dimension is $11 \times 257 = 2827$. There are 3 hidden layers, each with 2048 sigmoid hidden nodes. The output layer has 257 nodes, the same as the dimensionality of the spectrum vectors. In DNN2, we used 15 frames (i.e. about 150ms) as the input and 3072 nodes for hidden layers. The output layer size is $257 \times 3 = 771$ nodes, as both the static, delta, and acceleration of the spectrum are predicted. DNN3 is the same as DNN2 except that the input context is increased further to 19 frames. DNN4 and

Table 1. Detailed Results of the DS beamformer and the spectral subtraction on speech enhancement for channel type of 1ch.

Case	Simulated Rooms							Real
	CD		SRMR	LLR		SNR		SRMR
	mean	median	mean	mean	median	mean	median	mean
far1	2.62	2.35	4.76	0.4	0.37	7.75	9.54	3.96
far2	5.0	4.77	3.54	0.71	0.6	2.48	4.46	
far3	4.75	4.46	3.27	0.82	0.74	1.4	3.02	
near1	1.98	1.74	4.55	0.37	0.35	8.91	10.41	3.91
near2	4.41	3.94	4.04	0.41	0.38	4.78	8.12	
near3	4.17	3.79	3.87	0.65	0.59	3.16	5.78	
Avg.	3.82	3.51	4.0	0.56	0.51	4.75	6.89	3.94

Table 2. Detailed Results of the DS beamformer and the spectral subtraction on speech enhancement for channel type of 2ch.

Case	Simulated Rooms							Real
	CD		SRMR	LLR		SNR		SRMR
	mean	median	mean	mean	median	mean	median	mean
far1	2.41	2.14	4.9	0.38	0.35	8.32	9.89	4.1
far2	4.82	4.55	3.77	0.62	0.53	3.24	5.78	
far3	4.55	4.23	3.4	0.75	0.69	2.03	4.05	
near1	1.81	1.59	4.67	0.34	0.33	9.51	10.68	4.1
near2	4.06	3.55	4.24	0.39	0.31	6.08	9.94	
near3	3.87	3.45	4.03	0.59	0.53	3.92	6.81	
Avg.	3.59	3.25	4.17	0.52	0.46	5.52	7.86	4.1

DNN5 both uses MVDR (8 channel) processed signal as input and 11 frames as input and 2048 nodes for hidden layers. The output layer sizes are 257 and 771 for DNN3 and DNN4, respectively.

From the results in Table 4, all DNN speech enhancement systems improve the objective evaluation metrics significantly. In addition, the results of DNN4 and DNN5 show that DNN enhancement is complementary to the MVDR beamforming except for LLR. By comparing DNN4 and DNN5, we can see that it is slightly beneficial to predict the static and dynamic spectrograms simultaneously and then use the dynamic part of the predicted spectrogram to improve the static spectrogram in the post processing. However, this is only an ad-hoc way to improve the smoothness of the predicted spectrogram. In the future, we will try to enforce the relationship between static and dynamic spectrogram during the DNN training. By comparing DNN1 and DNN2, we found that using larger hidden layers and longer input context produces better objective evaluation scores for simulated rooms. However, for the real room, the SRMR becomes worse. This could be due to that larger DNN may leads to overfitting and degrades performance on test data not similar to the DNN training data. When the context size is further increased to 19 frames in DNN3, no obvious performance improvement can be observed, despite the fact that 19 frames span only about 190ms, which is much smaller than the reverberation time of the test data. It could be due to that current DNN training method is not able to make full use of longer context size for better dereverberation.

The detailed results of DNN5 (MVDR+DNN) is shown in Table 5. By comparing DNN5 results with the best DS beamformer + SS results in Table 3, we found that MVDR+DNN performs better for CD and SRMR measures, while DS+SS is better for LLR. For SNR, the results are mixed. The mean improvement of SNR

Table 3. Detailed Results of the DS beamformer and the spectral subtraction on speech enhancement for channel type of 8ch.

Case	Simulated Rooms							Real
	CD		SRMR	LLR		SNR		SRMR
	mean	median	mean	mean	median	mean	median	mean
far1	1.94	1.73	5.19	0.33	0.3	9.79	10.88	4.63
far2	4.23	3.86	4.43	0.48	0.42	4.76	8.06	
far3	3.93	3.56	3.83	0.62	0.57	3.64	6.2	
near1	1.52	1.35	4.96	0.29	0.27	10.59	11.35	4.6
near2	3.24	2.75	4.54	0.27	0.2	8.32	12.52	
near3	3.16	2.75	4.41	0.46	0.4	5.84	9.06	
Avg.	3.0	2.67	4.56	0.41	0.36	7.16	9.68	4.62

Table 4. Comparison of DNN-based speech enhancement settings. The results are averaged over near and far test cases.

DNN	Simulated Rooms							Real
	CD		SRMR	LLR		SNR		SRMR
	mean	median	mean	mean	median	mean	median	mean
DNN on single channel								
None	3.97	3.69	3.68	0.58	0.51	3.62	5.39	3.18
DNN1	2.64	2.41	5.78	0.52	0.48	7.19	8.09	4.54
DNN2	2.50	2.28	5.77	0.50	0.47	7.55	8.35	4.36
DNN3	2.50	2.28	5.83	0.50	0.47	7.52	8.34	4.39
MVDR (8-ch) + DNN								
MVDR	3.64	3.28	4.85	0.48	0.43	5.31	7.76	4.12
DNN4	2.28	2.07	5.88	0.47	0.44	8.44	8.88	4.51
DNN5	2.23	2.04	5.94	0.47	0.44	8.52	8.95	4.51

using MVDR+DNN is larger, while the median improvement using DS+SS is larger. This is because the DNN speech enhancement improves less for utterances that are already in good quality. Using the CPU clock function, the DNN speech enhancement (not including MVDR) run time is about 8 times of the reference code.

4.2. Speech Recognition

4.2.1. Clean condition training

We used the HTK-based ASR system from the organizer for evaluation on clean condition training. The features are the first 13 Mel-frequency cepstral coefficents (MFCC, c0-c12) with their first and second derivatives. Three levels of processing in signal, feature and model levels are applied. In the signal level, MVDR beamformer is applied when more than 1 microphone is available. In the feature level, the speech feature statistics is normalized by first using MVN and then by the proposed cross transform with 33 context size (i.e. $L = 16$), both applied in utterance-based mode. Finally, a 256-class CMLLR model adaptation is applied in full batch mode.

Detailed results are shown in Table 6. We have 3 observations. First, MVDR beamformer is very effective in improving ASR performance. If 8-channel microphone data is available, up to 16% absolute WER reduction (53.9% to 37.9%) can be obtained. Second, the performance of the cross transform feature adaptation and the CMLLR model adaptation are similar, despite that the cross transform uses only one utterance for parameter estimation while the CMLLR uses whole test sets. This is partially due to that CMLLR is only

Table 5. Detailed Results of DNN5 speech enhancement.

Case	Simulated Rooms							Real
	CD		SRMR	LLR		SNR		SRMR
	mean	median	mean	mean	median	mean	median	mean
far1	1.78	1.66	5.81	0.39	0.38	9.87	9.99	4.64
far2	2.73	2.47	6.10	0.55	0.51	7.27	7.78	-
far3	2.58	2.34	5.35	0.52	0.48	7.62	8.37	-
near1	1.68	1.56	5.95	0.38	0.37	9.89	9.98	4.37
near2	2.22	2.02	6.34	0.47	0.44	8.31	8.69	-
near3	2.38	2.17	6.11	0.48	0.44	8.16	8.87	-
Avg.	2.23	2.04	5.94	0.47	0.44	8.52	8.95	4.51

Table 6. ASR performance (WER) using clean condition training data on the evaluation data. CT stands for cross transform, while MA refers to the 256-class based CMLLR model adaptation.

CT	MA	Simulated Rooms						Real		Avg.
		Room1A		Room2A		Room3A		Room1		
		near	far	near	far	near	far	near	far	
Single Microphone										
N	N	19.0	25.6	34.5	69.8	47.1	78.3	80.2	76.6	53.9
Y	N	15.6	20.7	24.2	45.3	30.9	57.5	63.1	62.4	**40.0**
N	Y	14.1	17.9	21.3	45.1	28.3	59.5	66.4	65.9	39.8
Y	Y	14.5	18.2	21.2	38.8	26.8	50.3	57.3	58.0	**35.6**
2 Microphones, with MVDR										
N	N	18.0	23.3	27.7	59.8	40.1	71.2	75.1	73.7	48.6
Y	N	14.5	19.0	20.6	38.8	26.6	51.0	56.5	58.6	**35.7**
N	Y	13.5	17.0	18.9	36.8	24.5	51.4	58.8	59.3	35.0
Y	Y	13.7	17.4	18.3	33.4	23.3	45.2	51.2	53.1	**31.9**
8 Microphones, with MVDR										
N	N	17.0	21.3	23.6	40.3	30.5	53.2	59.3	58.1	37.9
Y	N	14.3	17.2	18.0	27.9	21.7	36.2	43.1	46.4	**28.1**
N	Y	13.6	16.4	17.3	26.6	20.1	35.6	44.4	46.1	27.5
Y	Y	13.7	16.2	15.8	24.1	19.5	32.3	38.1	42.6	**25.3**

adapted to test environment, but not test speakers in the full batch mode. On the other hand, cross transform is estimated in utterance mode and thus able to implicitly compensate for both speaker variations and reverberation. Finally, the cross transform works complementary with the CMLLR. This is mainly because the cross transform uses temporal information up to 33 frames (about 0.33s).

4.2.2. Multi condition training

The performance of our ASR system using multi condition training is shown in Table 7. When more than 1 microphone is available, MVDR beamforming is applied. The MFCC features are normalized by utterance-based MVN and then enhanced by DNN-based feature compensation. The DNN takes 15 frames of MFCC as input, hence the input layer dimension is $15 \times 39 = 585$. We empirically choose to use 3 hidden layers with 2048 hidden nodes in each layer. The output layer is 39 dimensional, as we are predicting the clean MFCC features. The feature compensation DNN is pretrained using RBM training and refined using MSE criterion.

DNN based acoustic model is built using the Kaldi toolkit [39]. The output layer of the DNN contains about 3500 classes, i.e. the number of tied triphone states. The input of the network is only 9 frames of MFCC, as using more context does not lead into much better results. We use 7 hidden layers and 2048 nodes per layer. The DNN acoustic model is first pretrained using RBM unsupervised training, then trained using cross entropy training, and finally refined by sequential MMI training, all using Kaldi's DNN recipe [39].

Table 7. WER obtained using multi condition training data on the evaluation data. MVDR beamformer is used when #mic>1.

#mic	Simulated Rooms						Real		Avg.
	Room1A		Room2A		Room3A		Room1		
	near	far	near	far	near	far	near	far	
No DNN Feature Compensation									
1	8.7	9.4	10.5	16.5	13.4	20.0	35.4	34.3	18.52
2	8.6	9.6	9.1	14.9	11.6	18.3	33.3	30.7	16.99
8	7.8	8.3	8.3	10.8	9.8	13.3	24.8	25.1	13.51
With DNN Feature Compensation									
1	8.9	8.8	8.8	13.9	11.4	15.5	32.2	32.7	16.51
2	8.5	8.6	7.9	12.4	10.1	14.8	29.1	29.1	15.08
8	7.5	8.2	7.4	9.7	8.9	11.3	22.7	24.4	12.50

From the results in Table 7, we have several observations. First, similar to clean condition training, the MVDR beamforming has a big impact on the performance. Second, with DNN feature compensation, the ASR performance is consistently improved. In addition, the performance gain is larger for simulated rooms than for real room, and larger for more reverberant cases (e.g. room3A-far) than for less reverberant cases (e.g. room1A-near). The results show that even when DNN is already used for acoustic modeling, it is still useful to use another DNN to compensate the features. We hypothesize two reasons for the usefulness of DNN feature compensation. One is that DNN feature compensation uses more information than DNN acoustic model, as both clean and multi condition data are used in its training. Another reason is that it may be useful to explicitly recover the clean features rather than let DNN acoustic model to automatically discover useful features for speech recognition.

5. CONCLUSIONS

For the speech enhancement task in the reverberation challenge 2014, we investigated both conventional beamforming methods and spectral subtraction and the DNN-based speech enhancement method. We found that after being trained from parallel clean and reverberant speech, the DNN is able to dereverberate speech signal effectively and works complementarily with beamforming technique. Similarly, we also train DNN to map reverberant features to clean features for the speech recognition task. Results show that DNN feature compensation improves recognition performance significantly even when the acoustic model is also based on DNN. This shows that it is beneficial to use a DNN to clean up the features first, rather than completely relying on the acoustic model DNN to learn discriminative and robust features for the recognition task. When the parallel clean and reverberant speech is not available, cross-transform that uses both spectral and temporal information can be used to reduce the mismatch between reverberant test features and the clean acoustic model.

An important question regarding the DNN based speech enhancement and feature compensation is that how much data do we need to train a universal speech enhancer? And will one big

DNN be able to handle different types of noise, SNR, reverberation time, speaker characteristics, and languages? We will move towards answering these questions in the future research.

6. REFERENCES

[1] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-13)*, 2013.

[2] T.H. Li, "Estimation and blind deconvolution of autoregressive systems with nonstationary binary inputs," *Journal Time Series Analysis*, vol. 14, no. 6, pp. 575–588, 1993.

[3] R. Chen and T.H. Li, "Blind restoration of linearly degraded discrete signals by gibbs sampling," *IEEE Trans. Signal Processing*, vol. 43, pp. 2410–2413, 1995.

[4] O. Cappe, A. Doucet, M. Lavielle, and E. Moulines, "Simulation-based methods for blind maximum-likelihood filter deconvolution," *IEEE Trans. Signal Processing*, vol. 73, no. 1, pp. 3–25, 1999.

[5] S. Gannot and M. Moonen, "Subspace methods for multimicrophone speech dereverberation," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 11, pp. 1074–1090, 2003.

[6] M. Triki and D.T.M. Slock, "Delay and predict equalization for blind speech dereverberation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2006, vol. 5, pp. 97–100.

[7] M. Delcroix, T. Hikichi, and M. Miyoshi, "Precise dereverberation using multichannel linear prediction," *IEEE Trans. Audio, Speech, Language Processing*, vol. 15, no. 2, pp. 430–440, 2006.

[8] S. Subramaniam, A. Petropulu, and C. Wendt, "Cepstrum-based deconvolution for speech dereverberation," *IEEE Trans. Speech Audio Processing*, vol. 4, no. 5, pp. 392–396, 1996.

[9] Berry D. Van Veen and Kevin M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, , no. 8, pp. 1–24, Oct. 1988.

[10] J. Allen and D. Berkley, "Multimicrophone signal processing technique to remove room reverberation from speech signals," *J. Acoust. Soc. Am.*, vol. 62, pp. 912–915, 1977.

[11] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Int. Conf. on Acoust. Speech and Sig. Proc.*, 1988, pp. 2578–2581.

[12] S. Fischer, "Beamforming microphone arrays for speech acquisition in noisy environments," *Speech Communication*, vol. 20, pp. 215–227, Dec 1996.

[13] E. Habets, J. Benesty, I. Cohen, S. Gannot, and J. Dmochowski, "New insights into MVDR beamformer in room acoustics," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 18, no. 1, pp. 158–170, Jan. 2010.

[14] E. Habets and J. Benesty, "A two stage beamforming approach for noise reduction and dereverberation," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 21, no. 5, pp. 945–958, May 2013.

[15] K. Lebart, J.M. Boucher, and P.N Denbigh, "A new method based on spectral subtraction for speech dereverberation," *ACUSTICA*, vol. 87, no. 3, pp. 359–366, May-Jun 2001.

[16] F.S. Pacheco and R. Seara, "Spectral subtraction for reverberation reduction applied to automatic speech recognition," in *Proc. of the fifth international telecommunications symposium (ITS2006) Fortaleza-CE, Brazil,*, Sep 2006, vol. 4, pp. 581–584.

[17] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAM0: a british english speech corpus for large vocabulary continuous speech recognition," in *Proc. ICASSP '95*, 1995, pp. 81–84.

[18] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language (HLT-91)*, 1992, pp. 357–362.

[19] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, "The multi-channel wall street journal audio visual corpus (MC-WSJ-AV): specification and initial experiments," in *Proc. ASRU '05*, 2005, pp. 357–362.

[20] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2008.

[21] T. H. Falk, C. Zheng, and W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1766–1774, 2010.

[22] A. Rix, M. Hollier, A. Hekstra, and J. G. Beerends, "Perceptual evaluation of speech quality (PESQ), the new ITU standard for end-to-end speech quality assessment, Part I-time-delay compensation," *J. Audio Eng. Soc.*, vol. 50, no. 10, pp. 755–764, 2002.

[23] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[24] H. W. Löllmann, E. Yilmaz, M. Jeub, and P. Vary, "An improved algorithm for blind reverberation time estimation," in *International Workshop on Acoustic Echo and Noise Control (IWAENC), Tel Aviv, Israel*, Aug 2010.

[25] O. L. Frost III, "An algorithm for linearly constrained adaptive array process.," *IEEE Proc.*, vol. 60, no. 8, pp. 926–935, Oct. 1972.

[26] E. A. Wan and A. T. Nelson, *Handbook of neural networks for speech processing*, chapter Networks for speech enhancement, Artech House, Boston, 1998.

[27] E. A. Wan and A. T. Nelson, *Foundations and Trends in Machine Learning*, chapter Learning deep architectures for AI, pp. 1–127, 2009.

[28] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, pp. 1527–1554, 2006.

[29] G. E. 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," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[30] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *to appear in IEEE Signal Processing Letters*, 2014.

[31] S. Furui, "Speaker independent isolated word recognizer using dynamic features of speech spectrum," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[32] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[33] Y. Li, H. Erdogan, Y. Gao, and E. Marcheret, "Incremental on-line feature space MLLR adaptation for telephony speech recognition," in *Proc. ICSLP '02*, Denver, USA, Sept. 2002, pp. 1417–1420.

[34] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[35] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 257–270, 2007.

[36] X. Xiao, E. S. Chng, and H. Li, "Normalization of the speech modulation spectra for robust speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1662–1674, Nov. 2008.

[37] X. Xiao, E. S. Chng, and H. Li, "Temporal filter design by minimum KL divergence criterion for robust speech recognition," in *Proc. ICASSP '13*, Vancouver, Canada, May.

[38] D. H. H. Nguyen, X. Xiao, E. S. Chng, and H. Li, "Generalization of temporal filter and linear transformation for robust speech recognition," in *submitted to ICASSP 2014*, 2014.

[39] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU '11*, Dec. 2011.